Discrete Mathematics

Lecture notes

Zsuzsanna Jankó

Introduction: There will be 2 lectures and 2 exercise classes (for 2 groups) every week.

Exam: written exam, 2 possible dates. The questions will be a mixture of topics from lectures, and problems that are similar to the problems discussed on the exercise class.

Homework:
All of the not yet discussed problems are homework.
One problem each week is for writing down and handing in. You can write the solution in English or German, by hand or with computer (preferably LaTeX).

All the other problems: you just have to mark if you solved it. The hand-in problem is worth more points than the regular ones.
To participate in the final exam, you are expected to collect 50% of the points in total.

Books:

- L. Lovász – J. Pelikán – K. Vesztergombi: Discrete Mathematics. Elementary and beyond.
  A lecture note version of this book is available here:
  `https://cims.nyu.edu/~regev/teaching/discrete_math_fall_2005/dmbook.pdf`
- J. Matoušek – J. Nešetřil: Invitation to Discrete Mathematics (Diskrete Mathematik - Eine Entdeckungsreise)

Lecture 1

(Lecture 1 and 2 are taken from the Matoušek-Nešetřil book)

Warm-up problems:

Problem 1. *There are 3 houses and 3 wells. Connect every house with every well, the roads are not allowed to cross. Is it possible?*

Problem 2. *There are 8 vertices. What is the maximal number of edges we can add such that there is no triangle?*

Relations

Definition 1. A relation is a set of ordered pairs. If X and Y are sets, any subset of the Cartesian product $X \times Y$ is called a relation between X and Y. The most important case is $X = Y$; then we speak of a relation on X, which is thus an arbitrary subset $R \subseteq X \times X$.

We use the notation xRy if $(x, y) \in R$

Example: relation $R = \{(1,2), (2,4), (3,2), (4,2), (4,4)\}$ on the set $\{1,2,3,4\}$.
How can we draw a relation: with an adjacency matrix or a directed graph.

Definition 2. The inverse relation R^{-1} to a given relation R is given by $R^{-1} = \{(y,x) : (x,y) \in R\}$. It arises by "reversing arrows" in R.
The symbol Δ_X denotes the smallest reflexive relation on a set X: $\Delta_X = \{(x,x) : x \in X\}$.

Definition 3. We say that a relation R on a set X is

- **reflexive** if xRx for every $x \in X$;
- **symmetric** if xRy implies yRx, for all $x, y \in X$;
- **antisymmetric** if, for every $x, y \in X$, xRy and yRx never hold simultaneously unless $x = y$;
- **transitive** if xRy and yRz imply xRz, for all $x, y, z \in X$.

Definition 4.

- A relation R on a set X is called an *equivalence* on X (or sometimes an *equivalence relation*) if it is reflexive, symmetric, and transitive. (Äquivalenzrelation)
- A relation R on a set X is called an *ordering* on X if it is reflexive, antisymmetric, and transitive. (eine Ordnung)
- A relation R on a set X is called a *linear ordering* on X if it is an ordering and moreover, for every two elements $x, y \in X$ we have xRy or yRx. (eine lineare Ordnung/ totale Ordnung)

Definition 5 (Composition of relations)**.** Let X, Y, Z be sets, let $R \subseteq X \times Y$ be a relation between X and Y , and let $S \subseteq Y \times Z$. be a relation between Y and Z. The composition of the relations R and S is the relation $T \subseteq X \times Z$. defined as follows: for given $x \in X$ and $z \in Z$, xTz holds if and only if there exists some $y \in Y$ such that xRy and ySz. The composition of relations R and S is usually denoted by $R \circ S$.

Note: it is the other way around with functions. The composition of functions f and g is usually denoted by $g \circ f$.

An equivalence relation defines eqivalence classses. Let R be an equivalence on a set X and let x be an element of X. By the symbol $R[x]$, we denote the set of all elements $y \in X$ that are equivalent to x;
$R[x] = \{y \in X : xRy\}$. $R[x]$ is called the equivalence class of R determined by x.

Proposition 1. *For any equivalence R on X, we have*
(i) $R[x]$ is nonempty for every $x \in X$.
(ii) For any two elements $x, y \in X$, either $R[x] = R[y]$ or $R[x] \cap R[y] = \emptyset$.
(iii) The equivalence classes determine the relation R uniquely.

Definition 6. $P = (X, \preceq)$ is called a partially ordered set (poset, for short) if $\preceq$ is an ordering on X.

Definition 7. Let $\preceq$ be an ordering. We say $a \prec b$ if $a \preceq b$ and $a \neq b$.

Drawing a poset: immediate predecessor

Definition 8. Let $(X, \preceq)$ be an ordered set. We say that an element $x \in X$ is an *immediate predecessor* of an element $y \in X$ if $x \prec y$ and there is no $t \in X$ such that $x \prec t \prec y$.

Let us denote the just-defined relation of immediate predecessor by $\triangleleft$

Proposition 2. *Let $(X, \preceq)$ be a finite ordered set, and let $\triangleleft$ be the corresponding immediate predecessor relation. Then for any two elements $x, y \in X$, $x \prec y$ holds if and only if there exist elements $x_1, x_2, ..., x_k \in X$ such that $x \triangleleft x_1 \triangleleft \dots x_k \triangleleft y$ (possibly with $k = 0$, i.e. we may also have $x \triangleleft y$).*

Definition 9. Let $(X, \preceq)$ be an ordered set. An element $a \in X$ is called a *minimal* element of $(X, \preceq)$ if there is no $b \in X$ such that $b \prec a$. A maximal element is defined analogously.

Definition 10. Let $(X, \preceq)$ be an ordered set. An element $a \in X$ is called a *smallest* element of $(X, \preceq)$ if for every $x \in X$ we have $a \preceq x$ (it is sometimes also called a minimum element). A *largest* element (sometimes also called a maximum element) is defined analogously.

Note: every smallest element is also a minimal element, but a minimal element does not have to be smallest.

Lemma 3. *Every finite partially ordered set $(X, \preceq)$ has at least one minimal element.*

Every linear ordering is also a (partial) ordering. The converse statement ("each partial ordering is linear") is obviously false. On the other hand, the following important theorem holds:

Theorem 4. *Let $(X, \preceq)$ be a finite partially ordered set. Then there exists a linear ordering $\leq'$ on X such that $x \preceq y$ implies $x \leq' y$.*

Each partial ordering can thus be extended to a linear ordering. The latter is called a *linear extension* of the former.

Lecture 2

Definition 11. (Important special types of functions). A function $f : X \to Y$ is called

- an injective function (one-to-one function) if $x \neq y$ implies $f(x) \neq f(y)$,
- a surjective function (or onto) if for every $y \in Y$ there exists $x \in X$ satisfying $f(x) = y$, and
- a bijective function, or bijection, if f is injective and surjective.

Example of a Poset: Let X be a set. The symbol 2^X denotes the system of all subsets of the set X. The relation "$\subseteq$" (to be a subset) defines a partial ordering on 2^X.

Definition 12. Let $(X, \preceq)$ and $(X', \preceq')$ be ordered sets. A mapping $f : X \to X'$ is called an embedding of $(X, \preceq)$ into $(X', \preceq')$ if the following conditions hold:
(i) f is an injective mapping;
(ii) $f(x) \preceq f(y)$ if and only if $x \preceq y$.

If f is an embedding that is also surjective, then it is an *isomorphism.* Isomorphism of ordered sets expresses the fact that they "look the same".

Theorem 5. *For every ordered set $(X, \preceq)$ there exists an embedding into the ordered set $(2^X, \subseteq)$.*

Proof. Proof. We show that, moreover, the embedding as in the theorem is very easy to find. We define a mapping $f : X \to 2^X$ by $f(x) = \{y \in X : y \preceq x\}$. We verify that this is indeed an embedding.

1. We check that f is injective. Let us assume that $f(x) = f(y)$. Since $x \in f(x)$ and $y \in f(y)$, the definition of f yields $x \preceq y$ as well as $y \preceq x$, and hence $x = y$ (by the antisymmetry of $\preceq$).
2. We show that if $x \preceq y$, then $f(x) \subseteq f(y)$. If $z \in f(x)$, then $z \preceq x$, and transitivity of $\preceq$ yields $z \preceq y$. The last expression means that $z \in f(y)$.
3. Finally, we show that if $f(x) \subseteq f(y)$, then $x \preceq y$. If $f(x) \subseteq f(y)$, then $x \in f(y)$, and hence $x \preceq y$.

□

Let $P = (X, \preceq)$ be a poset.

Definition 13. A set $A \subseteq X$ is called *independent* in P if we never have $x \preceq y$ for two distinct elements $x, y \in A$.
An independent set is also referred to as an *antichain.*

Let $\alpha(P)$ denote the maximum size of an independent set in P. In symbols, this can be written $\alpha(P) = max\{|A| : A \text{ is independent in } P\}$.
This $\alpha(P)$ can be thought of as a kind of abstract "width" of the ordered set P.

Observation 6. *The set of all minimal elements in P is independent.*

Proof. Suppose that x and y are minimal elements and $x \prec y$. Then y is not a minimal element. Contradiction. □

Definition 14. A set $C \subseteq X$ is called a chain in P if every two of its elements are comparable (in P).

Equivalently, the elements of C form a linearly ordered subset of P. Let $\omega(P)$ denote the maximum number of elements of a chain in P.
In other words, $\omega(P) = max\{|C| : C \text{ is a chain in } P\}$.
$\omega(P)$ can be thought of an the "height" of P.

Theorem 7 (Mirsky)**.** *For every finite partially ordered set, $\omega(P)$ equals the minimum number of antichains into which the set may be partitioned.*

Proof. In such a partition, every two elements of the longest chain must go into two different antichains, so the number of antichains is always greater than or equal to the height.

We want to show that $\omega(P)$ antichains are enough. For any $x \in X$, define $l(x)$ as the size of the longest chain whose greatest element is x. Define A_i as $A_i := \{x \in X : l(x) = i\}$.

$A_1 \cup \cdots \cup A_{\omega(P)}$ is a partition of X into $\omega(P)$ mutually disjoint sets.
We show that very A_i is an antichain. Suppose that A_i is not an antichain, exists two points x, $y \in A_i$ so that $x < y$. Take the longes chain to x, and add y. This is a chain of lenght $l(x) + 1$ whose greatest element is y. This implies $l(x) < l(y)$, contradiction.

□

Another formulation of Mirsky's theorem is that there always exists a partition for which the number of antichains equals the height.

Theorem 8. *For every finite ordered set $P = (X, \preceq)$, we have $\alpha(P) \cdot \omega(P) \geq |X|$.*

This is a corollary of Mirsky's theorem.

Theorem 9. *(Erdős–Szekeres lemma). An arbitrary sequence $(x_1, ..., x_{n^2+1})$ of real numbers contains a monotone subsequence of length $n + 1$.*

For example, the sequence $(3, 5, 6, 2, 8, 1, 4, 7)$ contains the monotone subsequence $(3, 5, 6, 8)$ or the monotone subsequence $(6, 2, 1)$ as well as many other monotone subsequences.

Proof. Let a sequence $(x_1, \ldots, x_{n^2+1})$ of $n^2 + 1$ real numbers be given. Let us put $X = \{1, 2, ..., n^2 + 1\}$, and let us define a relation $\preceq$ on X by $i \preceq j$ if and only if both $i \leq j$ and $x_i \leq x_j$. It is not difficult to verify that the relation $\preceq$ is a (partial) ordering of the set X. So we have $\alpha(X, \preceq) \cdot \omega(X, \preceq) \geq n^2 + 1$, and hence $\alpha(X, \preceq) > n$ or $\omega(X, \preceq) > n$. Now it is easily checked that a chain $i_1 \prec i_2 \prec \cdots \prec i_m$ in the ordering $\preceq$ corresponds to a nondecreasing subsequence $x_{i_1} \leq x_{i_1} \leq \cdots \leq x_{i_1}$ (note that $i_1 < i_2 < \cdots < i_m$), while an independent set $\{i_1, i_2, \ldots, i_m\}$ corresponds to a decreasing subsequence.

□

Theorem 10 (Dilworth)**.** *In a partially ordered set P, the size of the maximum antichain equals the minimum number k of chains such that P can be partitioned into k chains.*

(I did not prove this in the lecture)

Proof. Let P be a finite partially ordered set. The theorem holds trivially if P is empty. So, assume that P has at least one element, and let a be a maximal element of P.

By induction, we assume that for some integer k the partially ordered set $P' := P \setminus \{a\}$ can be covered by k disjoint chains $C_1, \ldots, C_k$ and has at least one antichain A_0 of size k. Clearly, $A_0 \cap C_i \neq \emptyset$ for $i = 1, 2, \ldots, k$. For $i = 1, 2, \ldots, k$, let x_i be the maximal element in C_i that belongs to an antichain of size $k \in P'$, and set $A := \{x_1, x_2, \ldots, x_k\}$. We claim that A is an antichain. Let A_i be an antichain of size k that contains x_i. Fix arbitrary distinct indices i and j. Then $A_i \cap C_j \neq \emptyset$. Let $y \in A_i \cap C_j$. Then $y \leq x_j$, by the definition of x_j. This implies that $x_i \not\geq x_j$, since $x_i \not\geq y$. By interchanging the roles of i and j in this argument we also have $x_j \not\geq x_i$. This verifies that A is an antichain.

We now return to P. Suppose first that $a \geq x_i$ for some $i \in \{1, 2, \ldots, k\}$. Let K be the chain $\{a\} \cup \{z \in C_i : z \leq x_i\}$. Then by the choice of x_i, $P \setminus K$ does not have an antichain of size k. Induction then implies that $P \setminus K$ can be covered by $k - 1$ disjoint chains since $A \setminus \{x_i\}$ is an antichain of size $k - 1$in $P \setminus K$. Thus, P can be covered by k disjoint chains, as required. Next, if $a \not\geq x_i$ for each $i \in \{1, 2, \ldots, k\}$, then $A \cup \{a\}$ is an antichain of size $k + 1$ in P (since a is maximal in P). Now P can be covered by the k+1 chains $\{a\}, C_1, C_2, \ldots, C_k$ completing the proof.

□

Lecture 3

Alice invites six guests to her birthday party: Bob, Carl, Diane, Eve, Frank and George. When they arrive, they all shake hands with each other.

Problem 3. *How many handshakes happened?*

Answer: There are 7 people and everyone shook hand with 6 people. This means $7 \cdot 6$ handshakes. But we counted every handshake twice. So, there were $\frac{7 \cdot 6}{2} = 21$ handshakes.

Problem 4. *How many ways can 6 people be seated at the table?*

The first person can choose from 6 places, the second from the remaining 5 places... In total, $6 \cdot 5 \cdot 4 \cdot 3 \cdot 3 \cdot 1 = 720$.

Problem 5. *At the party, Frank had an idea:*
"Let's pool our resources and win a lot on the lottery! All we have to do is to buy enough tickets so that no matter what they draw, we should have a ticket with the right numbers. How many tickets do we need for this?"
(In the lottery they are talking about, 5 numbers are selected from 90.)

Finally, the six guests decide to play chess. Alice, who just wants to watch them, sets up 3 boards.

Problem 6. *How many ways can the 6 guests be matched with each other? It does not count it as a different matching if two people at the same board switch places, and it should not matter which pair sits at which table.*

Definition 15. Let $n! = 1 \cdot 2 \cdot 3 \cdots n$. This is called n factorial. (in German: n Fakultät)

Note: $0!$ is defined as 1.

Definition 16. Set S has n elements. We denote the number of subsets of S of size k with $\binom{n}{k}$. This is called n choose k.

	English	German
$\binom{n}{k}$	n choose k	n über k
$\frac{n}{k}$	n over k	n (geteilt) durch k

Proposition 11. $\binom{n}{k} = \binom{n}{n-k}$

Proposition 12. $\binom{n}{k} = \frac{n!}{(n-k)!k!}$

Problem 7. *There are 12 students in a class. We want to give them 17 chocolates. Every student receives at least 1 chocolate. How many ways can we do it?*

Take 17 chocolates and place 11 separators between them. The first student receives the chocolates from the start to the first separator, the second student gets the ones from the first separator to the second. There are 16 places to place these separators, so we have $\binom{16}{11}$ options. If there are n students and m chocolates, $(m \geq n)$, the solution is $\binom{m-1}{n-1}$.

	Permutation	Variation	Combination
Without repetition	$n!$	$n \cdot (n-1) \cdots (n-k+1) = \frac{n!}{(n-k)!}$	$\binom{n}{k}$
With repetition	$\frac{n!}{n_1!n_2!\cdots n_k!}$	n^k	$\binom{n+k-1}{k}$

We illustrate these 6 problems with 6 questions.

	Permutation	Variation	Combination
Without repetition	Question 1	Question 2	Question 3
With repetition	Question 4	Question 5	Question 6

1. (a) *How many ways can 10 people stand in a line?*
   Answer: $10! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10 = 3628800$
   The first person can stand in 10 places, the second one in the remaining 9 places, the third one in 8 possible places and so on.

   (b) *How many ways can n people stand in a line?*
   The first person can stand in n places, the second one in the remaining $n-1$ places, and so on, in total there are $n!$ possibilities.

2. (a) *Bob wants to sent postcards to 3 of his friends. There are 6 kinds of postcards available, one from each type. How many ways can he do it?*
   Answer: $6 \cdot 5 \cdot 4 = 120$
   We choose from 6 possibilities for first postcard, 5 possibilities for the second postcard and 4 for the third postcard.

   (b) *Bob wants to sent postcards to k of his friends. There are n kinds of postcards available, one from each type. How many ways can he do it?*
   Answer: $n \cdot (n-1) \cdot (n-2) \cdots (n-k+1)$

3. (a) *I have 10 marbles. How many ways can I pick 4 of them?*
   Answer: $\binom{10}{4} = \frac{10!}{6!4!} = \frac{10 \cdot 9 \cdot 8 \cdot 7}{4!} = 210$

   (b) *I have n marbles. How many ways can I pick k of them?*
   Answer: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

4. (a) *How many 6 digit number can be formed using the digits $1, 1, 2, 3, 3, 3$?*
   Answer: If we had 6 different digits, the answer would be $6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$
   But we do not differentiate between the two 1s, their order does not matter. Divide by 2. We also not differentiate between the three 3s, (and they can be placed in $3! = 6$ different orders) divide by 6.
   In total, we have $\frac{6!}{2!3!} = \frac{720}{12} = 60$ such numbers.

   (b) *How many n-character word can be formed it we have k different characters and we can use the first character n_1 times, the second n_2 times, $\ldots$ the k^{th} character n_k times. We know that $n_1 + n_2 + n_3 + \ldots n_k = n$*
   Answer: $\frac{n!}{n_1! n_2! \cdots n_k!}$

5. (a) *Using the letters a, b, c, how many 4 letter words can be formed ? We are free to use any letter multiple times. (The word does not have to actually make sense.)*
   Answer: For each letter, we can choose any of the 3. So we have 3^4 options in total.

(b) *We have n different symbols. How many words of lenght k can be formed from them?*

Answer: For each of the k letters, we can choose any of the n symbols. So we have n^k options in total.

(c) *Bob wants to sent postcards to k of his friends. There are n kinds of postcards available, several (more than needed) from each type. How many ways can he do it?*

Answer: n^k

6. (a) *There are 12 students in a class. We want to give out 5 chocolates. One student may receive more than 1 chocolate. How many ways can we do it?*

Answer: Take 12 extra chocolates and rephase the problem: now we are giving them 17 chocolates, and at least one to everyone. This was solved in Problem 7, so the answer is $\binom{16}{11} = \binom{16}{5}$.

Second solution: Take the 5 chocolates and place 11 separators.

(b) *There are n students in a class. We want to give them k chocolates. One student may receive more than 1 chocolate. How many ways can we do it?*

Take the k chocolates and place $n-1$ separators. Now we do not need everyone to receive at least one chocolate, therefore we can put the $n-1$ separators anywhere. If two separators are right next to each other, the corresponding student receives nothing. So we have $k+n-1$ objects in total in a row (k chocolates and $n-1$ separators). Therefore we can place the separators in $\binom{n+k-1}{n-1} = \binom{n+k-1}{k}$ ways.

Problem 8. $x_1 + x_2 + \cdots + x_n = k$

$x_i \in \mathbb{Z}$ *and* $x_i \geq 0 \quad \forall i \in \{1, 2, \ldots n\}$

How many solutions does this equation have?

Answer: $\binom{n+k-1}{k}$

We can see that this is the same as the chocolate problem. We have k chocolates, and x_i denotes how many chocolates does the i^{th} student get.

Problem 9. $x_1 + x_2 + x_3 = 3$

$x_1, x_2, x_3 \in \mathbb{Z}$

$x_1, x_2, x_3 \geq 0$

How many solutions does this equation have?

Answer: $\binom{n+k-1}{k} = \binom{3+3-1}{3} = \binom{5}{3} = 10$

Problem 10. $x_1 + x_2 + x_3 = 8$

$x_1, x_2, x_3 \in \mathbb{Z}$

$x_1, x_2, x_3 \geq 0$

How many solutions does this equation have?

Answer: $\binom{n+k-1}{k} = \binom{3+8-1}{8} = \binom{10}{8} = \binom{10}{2} = 45$

Lecture 4

Postcard problem again: *Bob wants to sent postcards to k of his friends. There are n kinds of postcards available, several (more than needed) from each type. How many ways can he do it?*

Proposition 13. *Let N be an n-element set (it may also be empty, i.e. we admit $n = 0, 1, 2, ...$) and let M be an m-element set, $m \geq 1$. Then the number of all possible mappings $f : N \to M$ is mn.*

Proposition 14. *A set with n elements has 2^n subsets.*

Proposition 15. *Set S has n elements. Then the number of subsets with even number of elements is 2^{n-1}.*

Binomial coefficients

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{3} + \cdots + \binom{n}{n} = 2^n$$

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{3} - \binom{n}{4} \cdots \pm \binom{n}{n} = 0$$

Theorem 16 (Binomial theorem)**.**

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

For example, let us substitute $x = y = 1$. What do we get?
$x = 1, y = -1$. What do we get?

Pascal triangle

Create the following picture. Every number is the sum of the two numbers above it.

$n=0$							1						
$n=1$						1		1					
$n=2$					1		2		1				
$n=3$				1		3		3		1			
$n=4$			1		4		6		4		1		
$n=5$		1		5		10		10		5		1	
$n=6$	1		6		15		20		15		6		1

Proposition 17.

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Proof. We can prove this in an algebraic and also in a combinatoric way.
$\binom{n-1}{k-1} + \binom{n-1}{k} = \frac{(n-1)!}{(k-1)!(n-k)!} + \frac{(n-1)!}{(k)!(n-k-1)!} = \frac{(n-1)!k+(n-1)!(n-k)}{(k)!(n-k)!} = \frac{(n-1)!n}{(k)!(n-k)!}$
$= \frac{n!}{(k)!(n-k)!} = \binom{n}{k}$
Second proof. Let N be an n-element set, and $a \in N$ is an element of it. Select k elements, we can do this $\binom{n}{k}$ ways. We either select a or not. If we selected a, we can choose the other elements $\binom{n-1}{k-1}$ ways, if not, we can do select the other elements $\binom{n-1}{k-1}$ ways. Therefore, $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$.

□

Proposition 18. *The k^{th} element of n^{th} row in the Pascal triangle is $\binom{n}{k-1}$.*

In other words, the numbers in the Pascal triangle are the same as the binomial coefficients. This is a corollary of Proposition 17.

Proposition 19. *A "sock" in the Pascal triangle:*

$$\binom{r}{r} + \binom{r+1}{r} + \binom{r+2}{r} + \dots \binom{n}{r} = \binom{n+1}{r+1}$$

Proofs: Induction alone, or using the Pascal triangle.

Fibonacci numbers

A story about rabbits.
We can define the Fibonacci sequence with the following recursion: $F_0 = 0, F_1 = 1$, and $F_{n+1} = F_n + F_{n-1}$ for every $n \geq 1$.

The first few values are $F_0 = 0, F_1 = 1, F_2 = 1, F_3 = 2, F_4 = 3, F_5 = 5 \dots$

Problem 11. *A staircase has n steps. You walk up taking one or two steps at a time. How many ways can you go up?*

$n = 1 :$ 1 way
$n = 2 :$ 2 ways
$n = 3 :$ 3 ways

If the number of steps needed is G_n, then $G_n = F_{n+1}$.

Lecture 5

On the exersice class, we showed that for Fibonacci numbers: $F_0 + F_1 + F_2 + \dots + F_n = F_{n+2} - 1$.

Proof by induction.

The Fibonacci numbers are given by the formula

$$F_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right)^n.$$

Where can we find the Fibonacci numbers in the Pascal triangle?
Look at the diagonal

$n=0$							1						
$n=1$						1		1					
$n=2$					1		2		1				
$n=3$				**1**		3		3		1			
$n=4$			1		4		**6**		4		1		
$n=5$		1		5		10		10		**5**		1	
$n=6$	1		6		15		20		15		6		**1**

Looking at the bold numbers, the sum is $1+6+5+1=13$, which is a Fiboncci number. Now add the numbers on all the diagonals that are parallel to this one. Notice that they are all Fibonacci numbers.

Counting regions

Problem 12. *Let us draw n lines in the plane. These lines divide the plane into some number of regions. How many regions do we get?*

A set of lines in the plane such that no two are parallel and no three go through the same point is said to be *in general position.* We assume that the lines are in general position.

For a small number of lines, the answer is 1, 2, 4, 7, 11 ...

Lemma 20. *Rule: if we have a set of $n-1$ lines in the plane in general position, and add a new line (preserving general position), then the number of regions increases by n.*

This way we get $1+(1+2+3\dots n)=1+\frac{n(n+1)}{2}$

Theorem 21. *A set of n lines in general position in the plane divides the plane into $1+\frac{n(n+1)}{2}$ regions.*

Second proof: look at the "lowest" point of each region. $n+1$ at the bottom of the blackboard, and $\binom{n}{2}$ intersections.

This way we get $1+n+\binom{n}{2}=1+n+\frac{n(n-1)}{2}=1+\frac{n(n+1)}{2}$ regions.

Inclusion–exclusion principle (Siebformel)

In a class of 40, many students are collecting the pictures of their favorite bands. Eighteen students have a picture of Rammstein, 16 students have a picture of the Die Prinzen and 12 students have a picture of Die Toten Hosen. There are 7 students who have pictures of both Rammstein and Prinzen, 5 students who have pictures of both Rammstein and Toten Hosen, and 3 students who have pictures of both Prinzen and Toten Hosen. Finally, there are 2 students who possess pictures of all three groups.
Question: How many students in the class have no picture of any of the bands?

Our result is: $40-(18+16+12)+(7+5+3)-2=7$.
$|A\cup B\cup C|=|A|+|B|+|C|-|A\cap B|-|A\cap C|-|B\cap C|+|A\cap B\cap C|$.

Theorem 22. $A_1,A_2\dots A_n$ *are finite sets. Then*

$$\left|\bigcup_{i=1}^{n}A_i\right|=\sum_{i=1}^{n}|A_i|-\sum_{1\le i<j\le n}|A_i\cap A_j|+\sum_{1\le i<j<k\le n}|A_i\cap A_j\cap A_k|-\dots(-1)^{n-1}|A_1\cap A_1\cap\dots A_n|$$

Lecture 6 (April 23)

Proof for the Inclusion–exclusion principle: An element x is exacly in j subsets. How many times did we count it?

The story of swapped hats

Problem 13. *There were n people at a party. They all have their own hats. The light went out, and everyone just picked a random hat when going home. What is the probality that no one has their own hat?*

In other words: What is the number of fixed point free permutations?

A permutation π of the set $\{1, 2, \ldots, n\}$ is a function $\pi : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$, where $\pi(i)$ is the number of the hat returned to the ith gentleman. The question is, what is the probability of $\pi(i) \neq i$ holding for all $i \in \{1, 2, \ldots, n\}$? Call an index i with $\pi(i) = i$ a fixed point of the permutation π. So we ask: what is the probability that a randomly chosen permutation has no fixed point?

Each of the $n!$ possible permutations is equally probable, and so if we denote by $D(n)$ the number of permutations with no fixed point on an n-element set, the required probability equals $D(n)/n!$.

Using the inclusion–exclusion principle, we derive a formula for $D(n)$. We will actually count the "bad" permutations, i.e. those with at least one fixed point. Let S_n denote the set of all permutations of $\{1, 2, ..., n\}$, and for $i = 1, 2, ..., n$, we define $A_i = \{\pi \in S_n : \pi(i) = i\}$. The bad permutations are exactly those in the union of all the A_i.

It is easy to see that $|A_i| = (n-1)!$, because if $\pi(i) = i$ is fixed, we can choose an arbitrary permutation of the remaining $n-1$ numbers. Which permutations lie in $A_1 \cap A_2$? Just those with both 1 and 2 as fixed points (and the remaining numbers can be permuted arbitrarily), and so $|A_1 \cap A_2| = (n-2)!$. More generally, for arbitrary $i_1 < i_2 < \cdots < i_k$ we have $|A_{i_1} \cap A_{i_2} \cdots \cap A_{i_k}| = (n-k)!$, and substituting this into the inclusion–exclusion formula yields

$$|A_1 \cup \cdots \cup A_n| = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} (n-k)! = \sum_{k=1}^{n} (-1)^{k-1} \frac{n!}{k!}$$

We counted the bad permutations, so

$$D(n) = n! - |A_1 \cup \cdots \cup A_n| = n! - \frac{n!}{1!} + \frac{n!}{2!} - \cdots + (-1)^n \frac{n!}{n!}$$

$$D(n) = n! \left(1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!}\right)$$

The series in parentheses converges to $1/e$. Therefore the probability hat noone has his own hat converges to $1/e$.

Definition 17. The greatest common divisor (gcd) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers. For example, the gcd of 8 and 12 is 4.
Integers a and b are *relative primes*, if $gcd(a, b) = 1$.

Definition 18. Let n be a positive integer. Euler's function counts the positive integers up to a given integer n that are relatively prime to n.

$\varphi(n) = |\{1 \leq i \leq n : gcd(i, n) = 1\}|$.

Let n be a positive integer. How many numbers of $\{1, 2, 3 \ldots n\}$ are relative prime to n?

Theorem 23. *The prime factorisation of n is $n = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$. Then*

$$\varphi(n) = n \cdot \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$$

Proof: Using the inclusion-exclusion principle.

Catalan numbers

On a grid we walk from $(0,0)$ to (n,n) and we have to stay under the diagonal. How many ways can we do it? The number of possbilities is C_n.

The first few values: $C_0 = 1$, $C_1 = 1$, $C_2 = 2$, $C_3 = 5$, $C_4 = 14$, $C_5 = 42 \ldots$

Statement 24. $C_n = \binom{2n}{n} - \binom{2n}{n+1} = \frac{1}{n+1}\binom{2n}{n}$ *for every* $n \geq 0$

Idea of proof: The number of all walks from $(0,0)$ to (n,n) is $\binom{2n}{n}$, beacuse we take $2n$ steps in total, n steps going right, n steps going up.
The walks that are "bad" (do not stay under the diagonal) touch a second diagonal line going from $(0,1)$ to $(n-1,n)$. The "good" walks do not touch this second diagonal. For bad walks, find the first point where it touches the second diagonal, and mirror the remaining part of the walk. This way we get a walk from $(0,0)$ to $(n-1,n+1)$. The number of such walks are $\binom{2n}{n+1}$. We can show that there is a one-to-one correspondence between bad walks and walks from $(0,0)$ to $(n-1,n+1)$. Therefore the number of good walks is $\binom{2n}{n} - \binom{2n}{n+1}$.

Other use of the Catalan numbers: C_n counts the number of expressions containing n pairs of parentheses which are correctly matched. For example:

((()))   ()(())   ()()()   (())()   (()())

Lecture 7 (April 26)

The Catalan numbers can be calculated with the following recursion.

$C_0 = 1$ and $C_{n+1} = \sum_{i=0}^{n} C_i\, C_{n-i}$ for $n \geq 0$

Pigeonhole principle (Schubfachprinzip)

The birthday problem
The birthday problem asks, for a set of n randomly chosen people, what is the probability that some pair of them will have the same birthday? By the pigeonhole principle, if there are 367 people in the room, we know that there is at least one pair who share the same birthday, as there are only 366 possible birthdays to choose from (including February 29, if present).
The birthday "paradox" refers to the result that even if the group is as small as 23 individuals, the probability that there is a pair of people with the same birthday is still above 50%.

People in London
We can demonstrate there must be at least two people in London with the same number of hairs on their heads. It is reasonable to assume (as an upper bound) that no one has more than 1,000,000 hairs on their head (m = 1 million holes). The population of London is more than 8,000,000.

Sock-picking
Assume a drawer contains a mixture of black socks and blue socks, each of which can be worn on either foot, and that you are pulling a number of socks from the drawer without looking. What is the minimum number of pulled socks required to guarantee a pair of the

same color? Using the pigeonhole principle, to have at least one pair of the same color (m = 2 holes, one per color) using one pigeonhole per color, you need to pull only three socks from the drawer (n = 3 items). Either you have three of one color, or you have two of one color and one of the other.

Proposition 25. *There are m holes and at least $m + 1$ pigeons sit in them. Then there exists a hole with at least 2 pigeons.*
There are m holes and at least $km+1$ pigeons. Then there exists a hole with at least $k+1$ pigeons.

Graphs

Prove that at a party with 51 people, there is always a person who knows an even number of others. (We assume that acquaintance is mutual. There may be people who don't know each other. There may even be people who don't know anybody else — of course, such people know an even number of others, so the assertion is true if there is such a person.)

A graph consists of a set of *nodes* (or points, or *vertices*, all these names are in use), and some pairs of these (not necessarily all pairs) are connected by *edges*. It does not matter whether these edges are straight of curvy; all that is important is which pair of nodes they connect. The set of nodes of a graph G is usually denoted by V; the set of edges, by E. Thus we write $G = (V, E)$ to indicate that the graph G has node set V and edge set E.

The only thing that matters about an edge is the pair of nodes it connects; hence the edges can be considered as 2-element subsets of V . This means that the edge connecting nodes u and v is just the set $\{u, v\}$. We'll further simplify notation and denote this edge by uv.

Coming back to our problem, we see that we can represent the party by a graph very conveniently. Our concern is the number of people known by a given person. We can read this off the graph by counting the number of edges leaving a given node. This number is called the *degree* of the node. The degree of node v is denoted by $d(v)$.

In the language of graph theory, we want to prove: *if a graph has an odd number of nodes, then it has a node with even degree.*

Theorem 26. *The sum of degrees of all nodes in a graph is twice the number of edges.*

$\sum_{v\in V} d(v) = 2|E|$, therefore the sum of degrees is an even number. This implies the previous statement.

The following statements are all corollaries of the fact that the sum of the degrees is an even number:

Statement 27.

- *A finite graph has even number of nodes with odd degree.*
- *If a graph has an odd number of nodes, then it has an odd number of nodes with even degree.*
- *If a graph has an even number of nodes, then it has an even number of nodes with even degree.*

Let us get acquainted with some special kinds of graphs. The simplest graphs are the *empty graphs*, having any number of nodes but no edges. We get another very simple kind of graphs if we take n nodes and connect any two of them by an edge. Such a graph is called a *complete graph* (or a clique). A complete graph with n nodes has $\binom{n}{2}$ edges.

Let us draw n nodes in a row and connect the consecutive ones by an edge. This way we obtain a graph with $n-1$ edges, which is called a *path*. The first and last nodes in the row are called the *endpoints* of the path. If we also connect the last node to the first, we obtain a cycle (or circuit).

Take a sequence of nodes and edges $v_1, e_1, v_2, e_2, v_3, \dots v_k$ such that $e_i = v_i v_{i+1}$ for every $1 \le i \le k-1$.

$v_1, e_1, v_2, e_2, v_3, \dots v_k$ is a

- path, if nodes and edges are never repeated
- trail, if nodes can be repeated but the edges are all different.
- walk, if nodes and edges can be repeated
- cycle, if $v_1 = v_k$, but other than that none of the nodes or edges are repeated. (A cycle is closed path.)
- circuit, if $v_1 = v_k$, and the nodes and edges can be repeated. (A circuit is closed walk.)

Definition 19. A graph H is called a *subgraph* of a graph G if it can be obtained from G by deleting some of its edges and nodes (of course, if we delete a node we automatically delete all the edges that connect it to other nodes).

In other words: $H = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$, $E' \subseteq E$ and V' contains all the endpoints of the edges in E'.

A key notion in graph theory is that of a *connected graph*. It is heuristically clear what this should mean, but it is also easy to formulate the property as follows: a graph G is connected if every two nodes of the graph can be connected by a path in G. To be more precise: a graph G is connected if for every two nodes u and v, there exists a path with endpoints u and v that is a subgraph of G. 4 Let G be a graph that is not necessarily connected. G will have connected subgraphs; for example, the subgraph consisting of a single node (and no edge) is connected. A connected component H is a maximal subgraph that is connected; in other words, H is a connected component if it is connected but every other subgraph of G that contains H is disconnected.

Definition 20. A graph $G = (V, E)$ is called a *tree* if it is connected and contains no cycle as a subgraph.

The simplest tree has one node and no edges. The second simplest tree consists of two nodes connected by an edge.

Theorem 28. *(a) A graph G is a tree if and only if it is connected, but deleting any of its edges results in a disconnected graph.*
(b) A graph G is a tree if and only if it contains no cycles, but adding any new edge creates a cycle.

Lecture 8 (April 30)

Theorem 29. *Every tree with at least two nodes has at least two nodes of degree 1.*

How to build a tree:
A real tree grows by developing a new twig again and again. We show that graph-trees can be grown in the same way. To be more precise, consider the following procedure, which we call the Tree-growing Procedure:
- Start with a single node.
- Repeat the following any number of times: if you have any graph G, create a new node and connect it by a new edge to any node of G.

Theorem 30. *Every graph obtained by the Tree-growing Procedure is a tree, and every tree can be obtained this way.*

Theorem 31. *Every tree on n nodes has $n-1$ edges.*

Definition 21. An Eulerian walk (or Eulerian trail) is a trail in a finite graph which visits every edge exactly once.

Definition 22. An Eulerian circuit is an Eulerian walk which starts and ends on the same vertex.

They were first discussed by Leonhard Euler while solving the famous Seven Bridges of Königsberg problem in 1736.

Theorem 32. *(a) If a connected graph has more than two nodes with odd degree, then it has no Eulerian walk.*
(b) If a connected graph has exactly two nodes with odd degree, then it has an Eulerian walk. Every Eulerian walk must start at one of these and end at the other one.
(c) If a connected graph has no nodes with odd degree, then it has an Eulerian circuit. (Every Eulerian walk is closed.)

We should pay attention: If a graph has *isolated vertices* (vertices of degree 0) it can still contain an Eulerian walk or circuit.

Theorem 33. *If a graph G does not have isolated vertices, the following two statements are equivalent.*
(i) G is connected graph, and has no nodes with odd degree
(ii) G has an Eulerian circuit.

Lecture 9 (May 3)

We can take any tree, select any of its nodes, and call it a root. A tree with a specified root is called a *rooted tree.*

Let G be a rooted tree with root r. Given any node v different from r, we know that the tree contains a unique path connecting v to r. The node on this path next to v is called the "parent" of v. The other neighbors of v are called the "children" of v. The root r does not have a parent, but all its neighbors are called its children.

Now a basic geneological assertion: every node is the parent of its children.

Proof. Indeed, let v be any node and let u be one of its children. Consider the unique path P connecting v to r. The node cannot lie on P: it cannot be the first node after v, since then it wold be the parent of v, and not its child; and it cannot be a later node, since then going from v to u on the path P and then back to v on the edge uv we would traverse a cycle. But this implies that adding the node u and the edge uv to P we get a path connecting u to r. Since v is the first node on this path after u, it follows that v is the parent of u. (Is this argument valid when $v = r$? Check!)

□

We have seen that every node different from the root has exactly one parent. A node can have any number of children, including zero.

A node with degree 1 is called a *leaf*.

How many trees are there on n nodes?

Before attempting to answer this question, we have to clarify an important issue: when do we consider two trees different?

- We fix the set of nodes, and consider two trees the same if the same pairs of nodes are connected in each. (This is the position the twon people would take when they consider road construction plans.) In this case, it is advisable to give names to the nodes, so that we can distinguish them. It is convenient to use the numbers $0, 1, 2, ..., n-1$ as names (if the tree has n nodes). We express this by saying that the vertices of the tree are labeled by $0, 1, 2, ...n-1$. Interchanging the labels 2 and 4 (say) would yield a different *labelled* tree.
- We don't give names to the nodes, and consider two trees the same if we can rearrange the nodes of one so that we get the other tree. More exactly, we consider two trees the same (the mathematical term for this is isomorphic) if there exists a one-to-one correpondence between the nodes of the first tree and the nodes of the second tree so that two nodes in the first tree that are connected by an edge correspond to nodes in the second tree that are connected by an edge, and vice versa. If we speak about unlabelled trees, we mean that we don't distinguish isomorphic trees from each other. For example, all paths on n nodes are the same as *unlabelled* trees.

How many simple graphs are there on n nodes? If the nodes are labeled, $2^{\binom{n}{2}}$.

(If the nodes are unlabeled: we don't know the exact formula, but definitely less than $2^{\binom{n}{2}}$.)

Theorem 34 (Cayley's formula)**.** *For every positive integer n, the number of trees on n labeled vertices is n^{n-2}.*

Proof. We prove this with the help of the Prüfer code.

long Prüfer code: Two lines, both lines contain $n-1$ numbers. The nodes of a tree on n nodes are labelled $0, \ldots, n-1$. We consider it as a rooted graph with node 0 as a root. To get the Prüfer code, in every step, we take the leaf with the smallest non-0 label. Delete it from the tree, and write down the label of the leaf in the upper row, and the label of its *parent* (its only neighbor) in the lower row. In the end, only the node 0 remains.

Prüfer code: Take the lower line of the long Prüfer code. The last number has to be zero. Remove this zero.

Properties: 1) The length of the code is $n-2$, since in every step, we removed one edge, (a tree has $n-1$ edges) and we did not include the last 0.
2) If a node v has degree $d(v)$, then v appears in the code exactly $d(v)-1$ times.
3) The Prüfer code gives a bijection between labelled trees on n nodes, and sequences of lenght $n-2$, containing the numbers $0,\dots,n-1$.

Decoding: 1) Write a 0 to the end.
2) From left to right, recreate the top row: always choose the smallest number that does not appear *later* in the bottow row, and does not appear *earlier* in the top row.
3) When we have both rows, the edges of the tree are given by the pair of numbers in each coloumn.

Since the Prüfer code gives a bijection between labelled trees on n nodes, and sequences of lenght $n-2$, containing the numbers $0,\dots,n-1$, the number of trees on n labeled vertices is n^{n-2}.

$\square$

The number of unlabelled trees: Let T_n be the number of unlabelled trees on n vertices.

$$\frac{n^{n-2}}{n!} < T_n < 4^{n-1}$$

Lecture 10-11 (May 7 and 10)

Minimum cost spanning tree

Definition 23. A *subgraph* of a graph $G=(V,E)$ is another graph $H=(V',E')$ formed from a subset of the vertices and edges of G. So $V'\subseteq V$, $E'\subseteq E$. The vertex subset must include all endpoints of the edge subset, but may also include additional vertices.

A *spanning subgraph* is one that includes all vertices of the graph; an *induced subgraph* is one that includes all the edges whose endpoints belong to the vertex subset.

Definition 24. A *spanning tree* T of an undirected graph G is a subgraph that is a tree which includes all of the vertices of G.

In general, a graph may have several spanning trees.

A graph that is not connected does not contain a spanning tree.

Every connected graph has a spanning tree: start removing edges as long as we keep the connectedness, if we cannot remove any more edges, the remaining graph is a tree, using Theorem 28.

Theorem 35. *Graph G has n nodes. Any two of the following three statements imply the third one:*
(i) G is connected
(ii) G does not contain a cycle
(iii) G has $n-1$ edges.

Proof. We saw earlier that (i) and (ii) implies (iii), every tree has $n-1$ edges.

If G satisfies (i) and (iii), take a spanning tree T of G. The spanning tree has $n-1$ edges, thus $T = G$, G is a treee.

If G satisfies (i) and (iii), but it is not connected, add edges to the graph. The edges we add always connect two different connected components of G, in this way, we cannot create a cycle in the graph. We stop adding edges when the graph is connected, at this time it is a tree. A tree has $n-1$ edges, so we actually added zero edges.

□

Definition 25. Let $V_1 \subseteq V$. A *cut* in the graph a set of edges such that exactly one endpoint of the edge is in V_1. $Cut(V_1) = \{uv \in E | u \in V_1, v \notin V_1\}$

Let $G = (V, E)$ be an undirected graph with a cost function over the edges, $c : E \to \mathbb{R}$. If $H = (V', E')$ is a subgraph of G, we use the following notation for the cost of the subgraph: $c(H) = \sum_{e \in E'} c(e)$.

Finding the minimum cost spanning tree:
A country with a n towns wants to construct a new telephone network to connect all towns. Of course, they don't have to build a separate line between every pair of towns; but they do need to build a connected network; in our terms, this means that the graph of direct connections must form a connected graph.

We will use the following shorthand notations. If $G = (V, E)$ is a graph, e is an edge,
$G + e = (V, E \cup \{e\})$
$G - e = (V, E \setminus \{e\})$
$V(G)$ and $E(G)$ denotes the vertex set and edge set of graph G.

Kruskal's algorithm
List the edges in the increasing order of cost: $c(e_1) \le c(e_2) \le \dots c(e_m)$.
At the start of the algorithm $T = (V, \emptyset)$
For $i = 1...m$
If $T + e_i$ does not contain a cycle, let $T := T + e_i$.
(Otherwise skip that edge, and T is unchanged)

"Pessimist" algorithm
At the start of the algorithm $T = G = (V, E)$
For $i = m...1$ (decreasing order or cost)
If $T - e_i$ is connected let $T := T - e_i$.
(Otherwise skip that edge, and T is unchanged)

Prim's algorithm (Jarník's algorithm)
Let v_0 be an arbitrary vertex of graph G.
At the start of the algorithm $T = (v_0, \emptyset)$
Take an edge uv that is a minimum cost edge in $Cut(V(T))$, $(u \in V(T), v \notin V(T))$ and add it to the tree.
$T := (V(T) \cup \{v\}, E(T) \cup \{uv\})$
Repeat this step until all vertices are in the tree.

Theorem 36. *Kruskal's algorithm gives a minimum cost spanning tree.*

Suppose the Kruskal algorithm gives T, and there is a spanning tree H such that $c(H) < c(T)$.

Let us imagine the process of constructing T, and the step when we first pick an edge that is not an edge of H. Let e be this edge. If we add e to H, we get a cycle C. This cycle is not fully contained in T, so it has an edge f that is not an edge of T. If we add the edge e to H and then delete f, we get a (third) tree H'. (Why is H' a tree? We removed and edge of cycle C, so H' is still connected, and it has $n-1$ edges.) We want to show that H' is at most as expensive as H. This clearly means that e is at most as expensive as f. Suppose (by indirect argument) that f is cheaper than e.

Now comes a crucial question: Why didn't the optimistic government select f instead of e at this point in time? The only reason could be that f was ruled out because it would have formed a cycle C' with the edges of T already selected. But all these previously selected edges are edges of H, since we are inspecting the step when the first edge not in H was added to T. Since f itself is an edge of H, it follows that all edges of C' are edges of H, which is impossible, since H is a tree. This contradiction proves that f cannot be cheaper than e and hence H cannot be cheaper than H'.

So we replace H by this tree H' that is not more expensive. In addition, the new tree H' has the advantage that it coincides with T in more edges, since we deleted from H an edge not in T and added an edge in T. This implies that if H' is different from T and we repeat the same argument again and again, we get trees that are not more expensive than H, and coincide with T in more and more edges. Sooner of later we must end up with T itself, proving that T was no more expensive than H.

Lemma 37. *For some $V_1 \subseteq V$, e is a minimum cost edge in $Cut(V_1)$. T is a spanning tree and $e \notin T$. Then there exist another spanning tree H such that $c(H) \leq c(T)$ and $e \in H$.*

Proof. Add edge e to the spanning tree T, $T + e$ contains a cycle, let us call this cycle C. This cycle must contain an edge of $Cut(V_1)$ other than e, this edge is f. Since e is a minimum cost edge in the cut, $c(e) \leq c(f)$.

Let $H := T + e - f$. Why is it a tree? Every vertex is connected to at one of the endpoints of e via the edges of $T - e$, and $C - e$ connects the two endpoint of e with each other. Thus $T + e - f$ is connected and has the same number of edges as T. So H is a tree, $e \in H$ and $c(H) \leq c(T)$. □

Corollary 38. *For some $V_1 \subseteq V$, e is a minimum cost edge in $Cut(V_1)$. Then there exist a minimum cost spanning tree T such that $e \in T$.*

Lemma 39. *For some cycle C in graph G, e is a maximum cost edge in C. T is a spanning tree, $e \in T$. Then there exist another spanning tree H such that $c(H) \leq c(T)$) and $e \notin H$.*

Remove edge e from the tree, $T - e$ is not connected. Let V_1 be a connected component of $T - e$. Since T connected, e is an edge in $Cut(V_1)$. Cycle C has at least two edges in common with $Cut(V_1)$, take an edge $f \in C \cap Cut(V_1)$, $f \neq e$. Let $H := T - e + f$. Since e was a maximum cost edge in the cut, $c(f) \leq c(e)$. H is a tree, $e \notin H$ and $c(H) \leq c(T)$.

Corollary 40. *For some cycle C in graph G, e is a maximum cost edge in C. Then there exist a minimum cost spanning tree T such that $e \notin T$.*

Theorem 41. *Prim's algorithm gives a minimum cost spanning tree.*

Mixed algorithm

A government wants to connect cities with roads, (i. e. they want to build a spanning tree). Optimists and pessimists win in unpredictable order. This means that sometimes they build the cheapest line that does not create a cycle with those lines already constructed; sometimes they mark the most expensive lines "impossible" until they get to a line that cannot be marked impossible without disconnecting the network, and then they build it.

Here, we keep track of T and set N (N meaning "not in the tree") At the start of the algorithm $T = (V, \emptyset)$, $N = (V, \emptyset)$

We set all the edges as "not examined".

At any step, choose one of the following steps.

- Let e_i one of the cheapest not examined edges. If $T + e_i$ does not contain a cycle, let $T := T + e_i$. $N := N$. If $T + e_i$ contains a cycle, T is unchanged, $N := N + e_i$. Label e_i as examined.

- Let e_i one of the most expensive not examined edges. If $G - N - e_i$ is not connected, let $T := T + e_i$, $N := N$. If $G - N - e_i$ is connected let $N := N + e_i$, $T := T$. Label e_i as examined.

The algorithm ends when every edge is examined.

Theorem 42. *The mixed algorithm gives a minimum cost spanning tree.*

Suppose the "mixed" algorithm gives T, and there is a spanning tree H such that $c(H) < c(T)$.

Let us imagine the process of constructing T, and look at the first time where an edge is examined and T and H chooses differently. Call this edge is e_i.

There are two possibilities:

- Case 1: e_i was one of the cheapest not examined edges at that time. Let T_0 denote what T was in the algorithm at that time when we examined e_i. If $T_0 + e_i$ contains a cycle, then $H + e_i$ also contains a cycle, since until that point in the algorithm T and H were the same. This is not possible, since H is a tree, thus e_i was selected by the algorithm, $e_i \in T$, but $e_i \notin H$.

  If we add e_i to H, we get a cycle C. This cycle is not fully contained in T, so it has an edge f that is not an edge of T. If we add the edge e to H and then delete f, we get a tree $H' = H + e_i - f$. (Why is H' a tree? We removed and edge of cycle C, so H' is still connected, and it has $n - 1$ edges.)

  All the edges $e_1, e_2 \dots e_i$ were the same in H and T, and $f \in H \setminus T$ therefore $c(e_i) \leq c(f)$.

  We can improve H to $H' = H + e_i - f$, and $c(H') \leq c(H)$.

- Case 2: e_i was one of the most expensive not examined edges at that time. Until that point in the algorithm T and H are the same, so if by excluding e_i T could not be connected any more, the same is true for H. Thus e_i was refused by the algorithm, $e_i \notin T$, but $e_i \in H$.

$H - e_i$ is not connected. There is an edge in T connecting two different components of $H - e_i$, let this be f. $H' = H - e_i + f$ is a tree (cycle free and has $n - 1$ edges). All the edges $e_{i+1}, e_{i+2} \ldots e_m$ were the same in H and T, and $f \in T \setminus H$ therefore $c(f) \leq c(e_i)$. Thus $c(H') \leq c(H)$.

In both cases, we improved H. The total cost did not increase and H' has one more edge in common with T than H has. Repeating this, we will reach a tree having the exactly the same edges as T, therefore T is optimal.

Theorem 43. *The "pessimist" algorithm gives a minimum cost spanning tree.*

The pessimist algorithm is a special case of the mixed algorithm, so it gives a minimum cost spanning tree.

Lecture 12 (May 14)

Travelling salesman problem

Definition 26. A *Hamiltonian cycle* is a cycle that contains all nodes of a graph.

The Hamilton cycle problem is the problem of deciding whether or not a given graph has a Hamiltonian cycle. Hamiltonian cycles sound quite similar to Eulerian walks: Instead of requiring that every edge be used exactly once, we require that every node be used exactly once. But much less is known about them than about Eulerian walks. Euler told us how to decide whether a given graph has an Eulerian walk; but no efficient way is known to check whether a given graph has a Hamiltonian cycle, and no useful necessary and sufficient condition for the existence of a Hamiltonian cycle is known.

The travelling salesman problem (TSP) asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city and returns to the origin city?"

But we want to show at least one simple algorithm that, even though it does not give the best solution, never looses more than a factor of 2. We describe this algorithm in the case when the cost of an edge is just its length, but it would not make any difference to consider any other measure (like time, or the price of a ticket), at least as long as the costs $c(ij)$ satisfy the *triangle inequality*: $c(ij) + c(jk) \geq c(ik)$

Let $G = (V, c)$ be an instance of the travelling salesman problem. That is, G is a complete graph on the set V of vertices, and the function c assigns a nonnegative real weight to every edge of G. According to the triangle inequality, for every three vertices u, v, and w, it should be the case that $c(uv) + c(vw) \geq c(uw)$

Lemma 44. *Let $G = (V, c)$ be an instance of the travelling salesman problem. If T is a minimum cost spanning tree, and H is a Hamiltonian cycle then $c(T) \leq c(H)$.*

Proof. Omit an edge e of H. $H - e$ is a spanning tree, and a min cost spanning tree is cheaper that just any arbitrary spanning tree. The cost of any edge is nonnegative, therefore $c(T) \leq c(H - e) \leq c(H)$.

□

Definition 27. There is a problem where we want to minimize the cost, let us denote the minimum cost with OPT. An algorithm for this problem gives a solution with cost C. This algorithm is called an *α-approximation*, if $C \leq \alpha \cdot OPT$.

Tree Shortcut Algorithm
Find a cheapest tree connecting the nodes. We can use any of the algorithms discussed in the previous section for this. So we find the cheapest tree T, with total cost $c(T)$. Now, how does this tree help in finding a tour? One thing we can do is to walk around the tree just as we did when constructing the "planar code" of a tree (when calculating the number of unlabeled trees). This certainly gives a walk that goes through each node at least once, and returns to the starting point.

Of course, this walk may pass through some of the towns more than once. But this is good for us: We can make shortcuts. If the walk takes us from i to j to k, and we have seen j already, we can proceed directly from i to k. Doing such shortcuts as long as we can, we end up with a tour that goes through every town exactly once.

Statement 45. *The Tree Shortcut Algorithm gives a 2-approximation to the travelling salesman problem.*

Proof. Denote the Hamiltonian cycle we get by the shortcut algorithm H, and the minimum cost Hamiltonian cycle by H_{OPT}.

The circuit we created by walking around the tree contains every edge of tree T exactly twice, thus it has a total cost $2c(T)$. Since the costs satisfy the triangle inequality $c(ij) + c(jk) \geq c(ik)$, in each shortcut step, the total cost never incresased. Therefore $c(H) \leq 2c(T)$

By Lemma 44, $c(T) \leq c(H_{OPT})$, thus $c(H) \leq 2c(T) \leq 2c(H_{OPT})$, the algorithm is a 2-approximation indeed. □

Cristofides algorithm:
The algorithm can be described in pseudocode as follows.

1. Create a minimum spanning tree T of G.
2. Let O be the set of vertices with odd degree in T. By the handshaking lemma, O has an even number of vertices.
3. Find a minimum-weight perfect matching M in the induced subgraph given by the vertices from O.
4. Combine the edges of M and T to form a connected graph H in which each vertex has even degree. (H may contain parallel edges.)
5. Form an Eulerian circuit in H.
6. Make the circuit found in previous step into a Hamiltonian circuit by skipping repeated vertices (shortcutting).

Statement 46. *The Cristofides algorithm gives a 1.5-approximation to the travelling salesman problem.*

Proof. Denote the Hamiltonian cycle we get with the Cristofides algorithm by H_{CR}, and the minimum cost Hamiltonian cycle by H_{OPT}.

A minimum cost spanning tree is T, by Lemma 44, $c(T) \leq c(H_{OPT})$. Let $G[O]$ be the induced subgraph of G, induced by the vertices in O. Let H_O be the Hamilton-path in $G[O]$ we get by visiting the vertices in O in the same order as the cycle H_{OPT} does. H_O can be created from H_{OPT} using shortcuts, so from the triangle inequality, $c(H_O) \leq c(H_{OPT})$.

The set O has an even number of vertices. Choosing every second edge in H_O, we see that H_O is the union of two perfect machings in $G[O]$. M is minimum-weight perfect matching, therefore $2c(M) \leq c(H_{OPT})$. Thus $c(T)+c(M) \leq \frac{3}{2}c(H_{OPT})$. Using the triangle inequality again, $c(H_{CR}) \leq c(T) + c(M) \leq \frac{3}{2}c(H_{OPT})$.

□

Lecture 13 (May 17)

Matchings

At the prom, 300 students took part. They did not all know each other; in fact, every girl new exactly 50 boys and every boy new exactly 50 girls (we assume, as before, that acquaintance is mutual). We claim that they can all dance simultaneously (so that only pairs who know each other dance with each other).

Definition 28. A graph is *bipartite* if its nodes can be partitioned into two classes, say A and B so that every edge connects a node in A to a node in B. A perfect matching is a set of edges such that every node of the graph is incident with exactly one of them.

$V = A \cup B$. We will use the notation $G = (A, B; E)$ for bipartite graphs.

Definition 29. Two edges are *independent* if they do not have a common endpoint.
A *matching* is a set of independent edges.
A *perfect maching* is a matching that covers all the vertices of a graph. In other words, a perfect matching is a set of edges such that every node is incident with exactly one of them.

After this, we can formulate our problem in the language of graph theory as follows: we have a bipartite graph with 300 nodes, in which every node has degree 50. We want to prove that it contains a perfect matching. As before, it is good idea to generalize the assertion to any number of nodes. Let's be daring and guess that the numbers 300 and 50 play no role whatsoever. The only condition that matters is that all nodes have the same degree (and this is not 0). Thus we set out to prove the following theorem:

Theorem 47. *If every node of a bipartite graph has the same degree $d \geq 1$, then it contains a perfect matching.*

Theorem 48 (The Marriage Theorem)**.** *A bipartite graph has a perfect matching if and only if $|A| = |B|$ and and any for subset of (say) k nodes of A there are at least k nodes in B that are connected to one of them.*

Definition 30. For any $X \subseteq V$, $\Gamma(X)$ denotes the set of neighbors of X.
$\Gamma(X) = \{v \in V | \exists u \in X \text{ such that } \exists uv \in E\}$

In the following, we say a matching M *covers* set A, if every node of A is an endpoint of some edge in matching M. (We cound also say that the matching saturates A.)

Theorem 49 (Hall)**.** *In a bipartite graph $G = (A, B; E)$ there is a matching covering A if and only if for every $X \subseteq A$, $|\Gamma(X)| \geq |X|$.*

Idea of proof It is easy to see that if there is a matching covering A then for every $X \subseteq A$, $|\Gamma(X)| \geq |X|$. Just consider the pairs given by the matching for every $x \in X$, this gives at least $|X|$ neighbors.

For the other direction, use an algorithm. We start with the empty matching. ......

Let $\nu(G)$ denote size of the maximum matching in graph G. (In other words, the maximum number of independent edges in G.)

A *vertex cover* in a graph is a set of vertices that includes at least one endpoint of every edge, and a vertex cover is *minimum* if no other vertex cover has fewer vertices.

Let $\tau(G)$ be the size of the minimum vertex cover.

Theorem 50 (Kőnig). *In a bipartite graph G, $\nu(G) = \tau(G)$*

Definition 31. We color the edges of the graph in a way that edges having a common endpoint should have different colors. $\chi'(G)$ denotes the mininum number of colors needed for a edge-coloring of graph G. (This is called the chromatic index of the graph.)

Theorem 51. *In a bipartite graph G, if every degree is d, we can color the edges with d colors. In other words, $\chi'(G) = d$.*

Lecture 14 (May 21)

Let $\Delta(G)$ denote the maximum degree in graph G. $\Delta(G) = \max_{v \in V} d(v)$

Theorem 52 (Kőnig). *In a bipartite graph G, $\chi'(G) = \Delta(G)$.*

Note that in this theorem, the graph does not have to be simple.

Planar graphs

We will use the following notations:
K_n denotes the complete graph on n vertices.
$K_{a,b}$ denotes the complete bipartite graph with a and b vertices in each color class.

That is, take a bipartite graph $G = (A, B; E)$ where $|A| = a, |B| = b$, and every node in A is connected by an edge to every node in B.

So far we have been studying properties of graphs not related to their drawings, and the role of drawings was purely auxiliary. In this chapter the subject of analysis will be the drawing of graphs itself and we will mainly investigate graphs that can be drawn in the plane without edge crossings. Such graphs are called planar.

In order to introduce the notion of a drawing formally, we define an *arc* first: this is a subset α of the plane of the form $\alpha = \gamma([0,1]) = \gamma(x) : x \in [0,1]$, where $\gamma : [0,1] \to \mathbb{R}^2$ is an injective continuous map of the closed interval $[0,1]$ into the plane. The points $\gamma(0)$ and $\gamma(1)$ are called the endpoints of the arc α.

Definition 32. By a drawing of a graph $G = (V, E)$ we mean an assignment as follows: to every vertex v of the graph G, assign a point $b(v)$ of the plane, and to every edge $e = \{v, v'\} \in E$, assign an arc $\alpha(e)$ in the plane with endpoints $b(v)$ and $b(v')$. We assume that the mapping b is injective (different vertices are assigned distinct points in the plane), and no point of the form $b(v)$ lies on any of the arcs $\alpha(e)$ unless it is an endpoint of that arc. A graph together with some drawing is called a *topological graph.* A drawing of a graph G in which any two arcs corresponding to distinct edges either have no intersection or only share an endpoint is called a planar drawing. A graph G is *planar* if it has at least one planar drawing

Faces of a graph drawing. Let $G = (V, E)$ be a topological planar graph, i.e. a planar graph together with a given planar drawing. Consider the set of all points in the plane that lie on none of the arcs of the drawing. This set consists of finitely many connected regions (imagine that we cut the plane along the edges of the drawing)

Let us stress that faces are defined for a given planar drawing. Faces are usually not defined for a nonplanar drawing, and also we should not speak about faces for a planar graph without having a specific drawing in mind.

(A topological planar graph, i.e. a planar graph together with a given planar drawing can be also called a *plane graph.*)

Theorem 53 (Euler's formula)**.** *Let G = (V,E) be a connected planar graph, and let f be the number of faces of some planar drawing of G. We use the notations $v = |V|$ and $e = |E|$. Then we have $f + v = e + 2$*

In the following theorems, n denotes the number of the vertices is the graph, and e denotes the number fo the edges.

Theorem 54. *If G is planar, connected, simple, every country is triangle (even the infinite country) then $e = 3n - 6$*

Theorem 55. *A simple planar graph on n nodes has at most $3n - 6$ edges.*

Theorem 56. *K_5 is not planar.*

Lecture 15 (May 24)

Theorem 57. *If G is a planar, simple, bipartite graph, then $e \leq 2n - 4$.*

Theorem 58. *$K_{3,3}$ is not planar.*

A *subdivision* of a graph G is a graph resulting from the *subdivision of edges* in G. The subdivision of some edge e with endpoints $\{u, v\}$ yields a graph containing one new vertex w, and with an edge set replacing e by two new edges, $\{u, w\}$ and $\{w, v\}$.

Let us remark that a graph G is planar if and only if each subdivision of G is planar. This property can be used for a combinatorial characterization of planar graphs – a characterization purely in graph-theoretic notions, using no geometric notions at all.

Theorem 59 (Kuratowski)**.** *A graph G is planar if and only if it has no subgraph isomorphic to a subdivision of $K_{3,3}$ or to a subdivision of K_5.*

Dual of a planar graph

Definition 33. (Dual graph). Let G be a topological planar graph, i.e. a planar graph (V, E) with a fixed planar drawing. Let $\mathcal{F}$ denote the set of faces of G. We define a graph, possibly with loops and multiple edges, of the form $(\mathcal{F}, E, \varepsilon)$, where ε is defined by $\varepsilon(e) = \{F_i, F_j\}$ whenever the edge e is a common boundary of the faces F_i and F_j (we also permit $Fi = Fj$, in the case when the same face lies on both sides of a given edge). This graph $(\mathcal{F}, E, \varepsilon)$ is called the (geometric) dual of G, and it is denoted by G^*.

Example: If G is a cycle of lenght 3, the dual graph has 2 nodes and 3 parallel edges.

Definition 34. (Chromatic number of a graph). Let $G = (V, E)$ be a graph, and let k be a natural number. A mapping $c : V \to \{1, 2, \ldots, k\}$ is called a coloring of the graph G if $c(x) \neq c(y)$ holds for every edge $\{x, y\} \in E$. The chromatic number of G, denoted by $\chi(G)$, is the minimum k such that there exists a coloring $c : V(G) \to \{1, 2, \ldots, k\}$.

Theorem 60 (6 color theorem). *Any planar graph G satisfies $\chi(G) \leq 6$.*

The proof is easy.

Theorem 61 (5 color theorem). *Any planar graph G satisfies $\chi(G) \leq 5$.*

The proof needs a trick: 1-3 recoloring, 2-4 recoloring.

Theorem 62 (4 color theorem). *Any planar graph G satisfies $\chi(G) \leq 4$.*

This is a famous theorem, but we did not prove this, the proof requires checking several cases by computer.

Theorem 63. *If every node in a graph has degree at most d, then the graph can be colored with $d + 1$ colors. $\chi(G) \leq d + 1$.*

Note that a graph is 2-colorable if and only if it is bipartite.

Theorem 64. *A graph is 2-colorable if and only if it contains no odd cycle.*

Proof. We already know the "only if" part of this theorem. A bipartite graph does not contain odd cycles.

To prove the "if" part, suppose that our graph has no odd cycle. Pick any vertex a and color it black. Color all its neighbors white. Notice that there cannot be an edge connecting two neighbors of a, because this would give a triangle. Now color every uncolored neighbor of these white vertices black. We have to show that there is no edge between the black vertices: no edge goes between u and the new black vertices, since the new black vertices didn't belong to the neighbors of a; no edge can go between the new black vertices, because it would give a cycle of length 3 or 5. Continuing this procedure the same way, if our graph is connected, we'll end up with 2-coloring all vertices. It is easy to argue that there is no edge between two vertices of the same color: Suppose that this is not the case, so we have two adjacent vertices u and v colored black (say). The node u is adjacent to a node u_1 colored earlier (which is white); this in turn is adjacent to a node u_2 colored even earlier (which is black); etc. This way we can pick a path P from u that goes back all the way to the starting node. Similarly, we can pick a path Q from v to the starting node. Starting from v, let's follow Q back until it first hits P, and then follow P forward to u. This path forms a cycle with the edge uv. Since the nodes along the path alternate in color, but start and end with black, this cycle is odd, a contradiction.

If the graph is connected, we are done: We have colored all vertices. If our graph is not connected, we perform the same procedure in every component, and obviously, this will give a good 2-coloring of the whole graph.

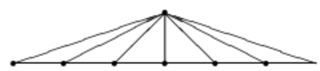
□

Lecture 16 (May 28)

Finite projective planes

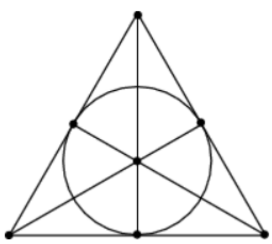
Let P be a finite set and let $\mathcal{L}$ be a system of subsets of P The elements of P are called points, the elements of $\mathcal{L}$ are called lines.

$(P, \mathcal{L})$ is a *finite projective plane* if it satisfies the following 3 axioms:
(i) Any two distinct sets $L_1, L_2 \in \mathcal{L}$ intersect in exactly one element, i.e. $|L_1 \cap L_2| = 1$.
(ii) For any two distinct elements $p_1, p_2 \in P$, there exists exactly one set $L \in \mathcal{L}$ such that $p_1 \in L$ and $p_2 \in L$.
(iii) There four points such that every line contains at most 2 of them. (Four points in general position.)



If we omit (iii) the so-called degenerate projective plane is also possible. Every point except one is on the same line. Here (i) and (ii) are satisfied, but we cannot find 4 points in general position.

Statement 65. *If $(P, \mathcal{L})$ is a finite projective plane, then every line contains at least 3 points.*



Let $d(p)$ denote the number of lines going thout point p, and $d(L)$ the number of points on line L.

Lemma 66. *If point p is not on line L, then $d(p) = d(L)$.*

Statement 67. *In a finite projective plane, there is an integer $n > 1$ such that any line contains $n+1$ points, and any point lies on $n+1$ lines.*

This n called the *order* of the projective plane.

Statement 68. *In a projective plane with order n, there are n^2+n+1 points and n^2+n+1 lines.*

An *affine plane* is a system of points and lines that satisfy the following axioms:
(i) Any two distinct points lie on a unique line.
(ii) Each line has at least two points.
(iii) Given any line and any point not on that line there is a unique line which contains the point and does not meet the given line.
(iv) There exist three non-collinear points (points not on a single line).

Lecture 17 (May 31)

The incidence graph of a finite projective plane is a bipartite graph $G = (A, B; E)$. $A =$ the nodes of the projective plane, $B =$ the lines of the projective plane. Point p and L are connected by an edge in the incidence graph if and only if p is on line L. ($p \in L$.)

If the order of the projective plane is n, the incidence graph has $2(n^2+n+1)$ vertices and it is $n+1$-regular.

Given a finite projective plane $(P, \mathcal{L})$, the dual of $(P, \mathcal{L})$ is obtained by taking the incidence graph of $(P, \mathcal{L})$ and interpreting as lines the vertices that were understood as points, and conversely, vertices that used to be sets start playing the role of points. Hence $\mathcal{L}$ is now thought of as a point set, and for each point $p \in P$, the set of lines $\{L \in \mathcal{L} : p \in L\}$ is interpreted as a line.

The dual of $(P, \mathcal{L})$ is a pair $(\mathcal{L}, \Lambda)$, where Λ is a system of subsets of $\mathcal{L}$, each of these subsets corresponding to some point of P. (Note that distinct points always yield distinct subsets of $\mathcal{L}$, since two points share only one line.)

Proposition 69. *The dual of a finite projective plane is also a finite projective plane.*

For which n does a projective plane with oder n exist?

Theorem 70 (Bruck–Ryser)**.** *If a finite projective plane of order n exists and n is congruent to 1 or 2 (mod 4), then n must be the sum of two squares.*

This implies that there is no projective plane with order 6 or 14.

Exist: $n =$ 2, 3, 4, 5, 7, 8, 9, 11, 13 ...
Does not exist: $n =$ 6, 10, 24 ...
Still an open question: $n =$ 12...

Construction, if the order is prime number.

Lecture 18 (June 4)

Proposition 71 (Cauchy–Schwarz inequality)**.** . *For arbitrary real numbers $x_1, x_2, \ldots, x_n$ and $y_1, y_2, ..., y_n$ we have*

$$\sum_{i=1}^{n} x_i y_i \leq \sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}$$

Theorem 72. *If a graph G on n vertices contains no subgraph isomorphic to $K_{2,2}$ then it has at most $\frac{1}{2}(n^{3/2} + n)$ edges.*

Idea of proof: counting cherries.

Proof. Let us write $V = V(G)$. We will double-count the size of the set M of all edge-pairs (so-called "cherries") $(\{u, u'\}, v)$, where $v, u, u' \in V$ and v is connected by an edge to both u and u'.

For a fixed pair $\{u, u'\}$, only one vertex $v \in V$ may exist joined to both u and u'. If there were two such vertices, v and v', they would together with u and u' form a subgraph isomorphic to $K_{2,2}$. Hence $|M| \leq \binom{n}{2}$. Now let us see how many elements of the form $(\{u, u'\}, v)$ are contributed to these M by a fixed vertex $v \in V$. For each pair $\{u, u'\}$ of its neighbors, v contributes one element of M, so if v has degree d it contributes $\binom{d}{2}$ elements Therefore, if we denote by $d_1, d_2, \ldots, d_n$ the degrees of the vertices of V, we obtain $|M| = \sum_{i=1}^{n} \binom{d_i}{2}$ Combining this with the previous estimate, we get

$$\sum_{i=1}^{n} \binom{d_i}{2} \leq \binom{n}{2}$$

We know that the number of edges of the graph is $\frac{1}{2}\sum_{i=1}^{n} d_i$. We can assume that our graph has no isolated vertices, and hence $d_i \geq 1$ for all i. Then we have $\binom{d_i}{2} \geq \frac{1}{2}(d_i - 1)^2$

$\sum_{i=1}^{n}(d_i - 1)^2 \leq \sum_{i=1}^{n} \binom{d_i}{2} \leq \binom{n}{2} \leq n^2$

Now we apply the Cauchy–Schwarz inequality with $x_i = d_i - 1$, $y_i = 1$.

We get

$$\sum_{i=1}^{n}(d_i - 1) \leq \sqrt{\sum_{i=1}^{n}(d_i - 1)^2}\sqrt{n} \leq \sqrt{n^2}\sqrt{n} \leq n^{3/2}$$

Therefore for the number of the edges $|E| = \frac{1}{2}\sum_{i=1}^{n} d_i \leq \frac{1}{2}(n^{3/2} + n)$

□

Using finite projective planes, we show that this bound is nearly the best possible in general:

Theorem 73. *For infinitely many values of n, there exists a $K_{2,2}$-free graph on n vertices with at least* $0.35\ n^{3/2}$ *edges.*

Block designs

The inhabitants of a town like to form clubs. They are socially very sensitive, and don't tolerate any inequalities. Therefore, they don't allow larger and smaller clubs (because they are afraid that larger clubs might suppress smaller ones). Furthermore, they don't allow some people to be members of more clubs than others, since those who are members of more clubs would have larger influence than the others. Finally, there is one further condition: Each citizen A must behave "equally" toward citizens B and C, A can not be in a tighter relationship with B than C. So A must meet B in the same number of clubs as he/she meets C.

We can formulate these strongly democratic conditions mathematically as follows. The town has v inhabitants; they organize b clubs; every club has the same number of members, say k; everybody belongs to exactly r clubs, and for any pair of citizens, there are exactly λ clubs where both of them are members.

The structure of clubs discussed in the previous paragraphs is called a *block design*. Such a structure consists of a set of v elements, together with a family of k-element subsets of this set (called blocks) in such a way that every element occurs in exactly r blocks, and every pair of elements occurs in λ blocks jointly. We denote the number of blocks by b. A degenerate case is when everyone is the member of the same club, that is $b = 1$, $k = v$. We will not consider this as a block design.

Examples:

- Every finite projective plane is a block design with $v = b = n^2 + n + 1$, $k + r + n + 1$ and $\lambda = 1$.
- Take a 3×3 grid where the block are the rows, coloums, the triples where the three points are in different rows and different coloums. This is a block design with $v = 9$, $b = 12$, $\lambda = 1$, $k = 3$, $r = 4$.
- Take all the size k subsets of a set V, $|V| = v$. Then $b = \binom{v}{k}$, $\lambda = \binom{v-2}{k-2}$ $r = \binom{v-1}{k-1}$

Statement 74. *For a block design, $bk = vr$.*

Statement 75. *For a block design, $\lambda(v-1) = r(k-1)$.*

Since we excluded the degenerate case, $k < v$, so $r > \lambda$.

Theorem 76 (Fisher inequality)**.** *For a block design, $b \geq v$.*

Proof. Let the incidence matrix M be a $v \times b$ matrix defined so that $M_{i,j}$ is 1 if element i is in block j and 0 otherwise. Then $B = MM^T$ is a $v \times v$ matrix such that $B_{i,i} = r$ and $B_{i,j} = \lambda$ for $i \neq j$.
Add each other row to row 1 of MM^T. Subtract column 1 in the resulting matrix from each other column. The result is the matrix shown here whose determinant is the product of its diagonal elements, which is $(r + (v-1)\lambda)(r - \lambda)^{v-1}$

$$\det(B) = \begin{vmatrix} r & \lambda & \lambda & \dots & \lambda \\ \lambda & r & \lambda & \dots & \lambda \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda & \lambda & \lambda & \dots & r \end{vmatrix} = \begin{vmatrix} r + (v-1)\lambda & 0 & 0 & \dots & 0 \\ \lambda & r - \lambda & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda & 0 & 0 & \dots & r - \lambda \end{vmatrix}$$

Since $r \neq \lambda$, $det(B) \neq 0$, so $rank(B) = v$; on the other hand, $rank(B) \leq rank(M) \leq b$, so $v \leq b$.

□

Lecture 19-20

Probability and probabilistic proofs

Let X be a finite set and let $\mathcal{M}$ be a system of subsets of X. Suppose that each set in $\mathcal{M}$ has exactly k elements. A 2-coloring a set-system means we color the elements with 2 colors in a way that none of the sets in $\mathcal{M}$ is monochromatic. Let $m(k)$ be the smallest number of sets in a system $\mathcal{M}$ that is not 2-colorable.

It is easy to find out that m(2) = 3, since we need 3 edges to make a graph that is not bipartite.

On the exercise class, we met a system of 7 triples that is not 2-colorable, namely the Fano plane, and so $m(3) \leq 7$.

Theorem 77. *We have $m(k) \geq 2^{k-1}$, i.e. any system consisting of fewer than 2^{k-1} sets of size k admits a 2-coloring.*

Theorem 78. $m(3) \geq 7$

Lemma 79. *Let X be a set with at most 6 elements, and let $\mathcal{M}$ be a system of at most 6 triples on X. Then $\mathcal{M}$ is 2-colorable.*

Problem 14. *Prove that $m(4) \geq 15$, i.e. that any system of 14 4-tuples can be 2-colored*

Lecture 21 (June 21)

A directed graph (shortly digraph) D is intuitively a graph together with an orientation of the edges, i.e. if e is an edge between vertices u and v then the orientation says that either e goes from u to v or e goes from v to u. More precisely D consists of a vertex set $V(D)$ and edge set (also called arc set) $A(D)$ and two functions $\mathsf{tail}_D, \mathsf{head}_D : A(D) \to V(D)$. We write simply V and A if D is clear from the context. We say that $e \in A(D)$ goes from u to v in D if $\mathsf{tail}_D(e) = u$ and $\mathsf{head}_D(e) = v$ and we write $e \in [u, v]_D$. An $e \in A(D)$ is a **loop** in D if $\mathsf{head}_D(e) = \mathsf{head}_D(e)$. The edges $e, f \in A(D)$ are **parallel** if $\mathsf{tail}_D(e) = \mathsf{tail}_D(f)$ and $\mathsf{head}_D(e) = \mathsf{head}_D(f)$. A (directed) **walk** from u to v in D is a sequence $v_0, e_0, v_1, e_1, v_1, \ldots, v_{n-1}, e_{n-1}, v_n$ such that $v_0 = u,\ v_n = v$, and $e_i \in A(D)$ goes from v_i to v_{i+1} $(0 \leq i \leq n)$. If in addition the e_i are pairwise distinct then it is a (directed) **tour**, furthermore, if the vertices v_i are pairwise distinct then we call it a (directed) **path**.

Problem 15. *Suppose that there is a given finite digraph D together with $s, t \in V$ and a cost function $c : A \to \mathbb{R}_+$. For a walk W, let us define the cost $\mathbf{c(W)}$ of W as the sum of the costs of the edges of W. We want to calculate the cost of a cheapest $s \to t$ path (if there is any) in an "efficient" way.*

Algorithm 80 (Dijkstra)**.** *The algorithm calculates the minimal cost $m(v)$ of the $s \to v$ paths in D for every $v \in V$ for which such a path exists. At the beginning let $m(s) = 0$ and let $m(v)$ be undefined for $v \in V \setminus \{s\}$. In a general step, consider the set S of those vertices v for which $m(v)$ has been already defined. If $\mathsf{out}_D(S) = \varnothing$ (where $\mathsf{out}_D(S)$ is the set of edges f with $\mathsf{tail}(f) \in S$ and $\mathsf{head}(f) \notin S$) then we stop. Otherwise we calculate $\alpha := \min\{m(u) + c(e) : u \in S, e \in \mathsf{out}_D(S), \mathsf{tail}(e) = u\}$ and take an edge e where the minimum is taken. For $v = \mathsf{head}(e)$, we define $m(v)$ to be α.*

Question 81. *How to modify the algorithm above to get also an $s \to v$ path P_v with $c(P_v) = m(v)$ for every v which is reachable from s?*

Problem 16. *Show by an example that Dijkstra's algorithm does not work if we allow negative edge costs.*

Let D be a digraph and $c : A \to \mathbb{R}$. A **potential** is a function $\pi : V \to \mathbb{R}$. We say that π is **feasible** (with respect to c) if $\pi(v) - \pi(u) \leq c(e)$ for every $e \in [u, v]_D$. A directed cycle is a cycle where the edges are oriented in such a way that every vertex is reachable by a directed path from any other vertex. For a directed cycle C, we define $c(C)$ to be the sum of the costs of its edges and we say that C is negative if $c(C) < 0$.

Definition 35. A cost function $c : A \to \mathbb{R}$ is called **conservative** if there is no negative cost directed cycle.

Let π be a feasible potential for cost funtion c. We will call an edge uv an *tight edge* if $\pi(v) - \pi(u) = c(e)$.

For any path P' with vertices $s = v_0, v_1, v_2 \dots t = v_k$,
$c(P') = \sum_{i=0}^{k-1} c(v_i v_{i+1}) \geq \sum_{i=0}^{k-1} (\pi(v_{i+1}) - \pi(v_i)) = \pi(t) - \pi(s)$ If P' is a cheapest path, all of its edges are tight edges. Therefore every beautiful edge is tight, and if we build a path from tight edges, it will be a cheapest path.

Lemma 82. *Let π be a feasible potential for cost function c. For every st-path P, $c(P) \geq \pi(t) - \pi(s)$. If every edge of the path is tight, then $c(P) = \pi(t) - \pi(s)$.*

For every directed cycle K, $c(K) \geq 0$. If every edge of the cycle is tight, then $c(K) = 0$.

Proof. For any path P with vertices $s = v_0, v_1, v_2 \dots t = v_k$,
$c(P) = \sum_{i=0}^{k-1} c(v_i v_{i+1}) \geq \sum_{i=0}^{k-1} (\pi(v_{i+1}) - \pi(v_i)) = \pi(t) - \pi(s)$. If the edges of P are tight edges, we have equality everywhere.

□

Theorem 83 (Gallai)**.** *In a finite digraph, there is a feasible potential if and only if there is no negative directed cycle.*

Proof. If there is a feasible potential, then for every cycle C with vertices $v_1 \dots v_k$, the total weight of the cycle is $c(C) = \sum_{i=i}^{k-1} c(v_i v_{i+1}) + c(v_k v_1) \geq \sum_{i=1}^{k-1} (\pi(v_{i+1}) - \pi(v_i)) + \pi(v_1) - \pi(v_k) = 0$. therefore there is no directed negative cycle.

Now, suppose that there is no negative directed cycle. Let π_c be a potential on the vertices: $\pi_c(v) :=$ the weight of the cheapest path ending in v. Since every path has at most $n - 1$ edges, there are finitely many paths ending in v, we can select the minimum cost path.

Note that since the cost of the path $\{v\}$, consisting of only one vertex, is 0, the function π_c is nonpositive.

We want to show that π_c is a feasible potential for c. Let uv be an edge of the digraph, and P_u is a cheapest path ending in u. If v is not on path P_u, then $P_v : +P_u + uv$ is a path ending in v, therefore $c(P_v) \geq \pi_c(v)$. Thus, $\pi_c(v) \leq c(P_v) = c(P_v) + c(uv) = \pi_c(v) + c(uv)$ so $\pi_c(v) - \pi_c(u) \leq c(uv)$.

If v is on path P_u, let P_1 denote the subpath of P_u from the beginning to v, and P_2 is the part of the path from v to u. Then $c(P_v) = c(P_1) + c(P_2)$. Let K be the $P_2 + uv$ directed cycle. Since c is conservative, $c(K) \geq 0$, thus $c(uv) \geq -c(P_2)$. Since P_1 is a path ending in v, $\pi_c(v) \leq c(P_1)$. Combining these, $\pi_c(v) - \pi_c(u) \leq c(P_1) - c(P_u) = -c(P_2) \leq c(uv)$. So π_c is indeed a feasible potential. □

Note: if c is integer valued, then π_c is also integer valued.

Let $\mu_c(v)$ be defined as $\mu_c(v) = \min\{c(P) : P \text{ is an } s \to v \text{ path}\}$. We want to show that this is a feasible potential. We show it similarly to the proof of Gallai's theorem.

Let uv be an edge of the digraph, and P_u is a cheapest path from s to u. If v is not on path P_u, then $P_v := P_u + uv$ is an sv path , therefore $c(P_v) \geq \pi_c(v)$. Thus, $\pi_c(v) \leq c(P_v) = c(P_v) + c(uv) = \pi_c(v) + c(uv)$ so $\pi_c(v) - \pi_c(u) \leq c(uv)$.

If v is on path P_u, let P_1 denote the subpath of P_u from the beginning to v, and P_2 is the part of the path from v to u. Then $c(P_v) = c(P_1) + c(P_2)$. Let K be the $P_2 + uv$ directed cycle. Since c is conservative, $c(K) \geq 0$, thus $c(uv) \geq -c(P_2)$. Since P_1 is a sv path, $\pi_c(v) \leq c(P_1)$. Combining these, $\mu_c(v) - \mu_c(u) \leq c(P_1) - c(P_u) = -c(P_2) \leq c(uv)$. So μ_c is indeed a feasible potential.

Note: if c is integer valued, then μ_c is also integer valued. $\mu_c(s) = 0$.

Theorem 84 (Duffin). *Let D be a finite digraph and $c : A \to \mathbb{R}$ such that there is no negative cycle. Then*

$$\min\{c(P) : P \textit{ is an } s \to t \textit{ path}\} = \max\{\pi(t) - \pi(s) : \pi \textit{ is a feasible potential}\}$$

(where we consider $\min \varnothing = +\infty$*).*

Proof. From Lemma 82, $c(P) \geq \pi(t) - \pi(s)$ for any path any feasible potential, therefore

$$\min\{c(P) : P \text{ is an } s \to t \text{ path}\} \geq \max\{\pi(t) - \pi(s) : \pi \text{ is a feasible potential}\}$$

We defined μ_c earlier, which is a feasible potential, and for μ_c

$$\min\{c(P) : P \text{ is an } s \to t \text{ path}\} = \mu_c(t) - \mu_c(s)$$

Therefore we have equality is hte statement of the Duffin theorem.

□

Note: Moreover, μ_c is the biggest feasible potential among the feasible potentials where $\mu(s) = 0$

Problem 17. *Let $D = (V, A)$ and $c : A \to \mathbb{R}$ and assume that there is no negative directed cycle. Give an algorithm that finds a a cheapest $s \to t$ path for some given $s, t \in V$. (Hint: if π is a feasible potential then the cost function d defined as $d(uv) := c(uv) - (\pi(v) - \pi(u))$ is non-negative and $\tilde{d}(P) - \tilde{c}(P) = \pi(s) - \pi(t)$ holds for every $s \to t$ path P. Combine Theorem 83 with Algorithm 80.)*

The Bellman-Ford algorithm is used to fint the cheapest path ending in v for any vertex v. The main difference between this algorithm and Dijkstra's the algorithm is that in Dijkstra's algorithm we cannot handle the negative weight, but here we can handle it.

$D = (V, A)$ and $c : A \to \mathbb{R}$. Now, we do not need to assume that c is conservative. For every $i = 0, 1, \ldots n$ let $W_c^{(i)}(v)$ be a cheapest walk with at most i edges ending in v. And $\pi_c^{(i)}(v)$ is the cost of the cheapest walk with at most i edges ending in v.

$\pi_c^{(0)}(v) = 0$ for every $v \in V$

After we calculated these up to i for every vertex,

$$\pi_c^{(i+1)}(v) = \min\{\pi_c^{(i)}(v), \min\{\pi_c^{(i)}(u) + c(uv) : uv \in A\}\}$$

If c is conservative, the cheapest walk with at most n edges is also the cheapest path ending in that given vertex.

If c is not conservative, the algorithm will find a negative cost cycle.

......

Lecture 22 (June 25)

Definition 36. Let V be the set of vertices of a graph/digraph, $s, t \in V$. An $s\bar{t}$ *set* is subset S of the V such that $s \in S$ but $t \notin S$.

In a directed graph, the indegree $d^-(v)$ of vertex v is the number of edges with head v. The outdegree $d^+(v)$ of vertex v is the number of edges with tail v.

For a vertex-set $S \subset V$, the indegree $d^-(S)$ of vertex S is the number of edges with entering S, i.e. the number of uv edges such that $u \notin S$, $v \in S$.
The outdegree $d^+(S)$ of vertex S is the number of edges with leaving S, i.e. the number of uv edges such that $u \in S$, $v \notin S$.

The directed paths P and Q are edge-disjoint if they have no common edges (but they might have common vertices).

Theorem 85 (Menger (directed, edge-version))**.** *Let D be a finite digraph and $s \neq t \in V$. There are k ($k \geq 1$) pairwise edge-disjoint directed $s \to t$ paths in the graph if and only if every $s\bar{t}$ set has an outdegree at least k.*

Proof. One direction:
If there are k pairwise edge-disjoint directed $s \to t$ paths in the graph, for any $s\bar{t}$ set S, all of these k paths have to use an arc leaving S, and they are using different arcs, therefore $d^+(S) \geq k$.

The other direction:
Suppose every $s\bar{t}$ set has an outdegree at least k. We use induction on the number of arcs. $|A| = 0$ is trivial. If there is a $s \to t$ path of lenght 1 or 2, remove this path. In the remaining graph, the outdegree of every $s \to t$ set decreased by exactly 1, therefore $d^+(S) \geq k - 1$ for every $s \to t$ set now. Using the induction hypothesis, there are $k - 1$ paths. Readd the path that we removed.

If every a st path has lenght at least 3, we can select an uv arc in a path such that $s \neq u$ and $v \neq t$. (Here we use there exist at least one $s \to t$ path. If there were none, we would wind a $s\bar{t}$ set with outdegree 0.) If we can remove arc uv and still every $s\bar{t}$ set has an outdegree at least k, remove the arc and use the induction hypothesis. We get k edge-disjoint paths.

If we cannot remove arc uv without ruining the condition, there is a $s\bar{t}$ set S such that $d^+(S) = k$, $u \in S$, $v \notin S$. Contract the set S to one point s. In the graph we get this way, we can check that still $s\bar{t}$ set has an outdegree at least k. The number of edges decreased, since the path containing uv had some edges between s and u. Using the induction hypothesis for the contracted graph, we find k edge-disjoint paths.

Do the same for $V \setminus S$ as well. Contract the set $V \setminus S$ to one point t. In the contracted graph, we can find k edge-disjoint paths. Since S had exactly k outgoing edges, both path sytems have to use these k edges. Glue the two path systems together at these k edges, this way we get k edge-disjoint paths for the original graph.

□

An alternative statement of the same theorem:

Theorem 86 (Menger (directed, edge-version)). *Let D be a finite digraph and $s \neq t \in V$. The maximal number of pairwise edge-disjoint directed $s \to t$ paths is equal to the minimal number of edges that we need to delete to destroy every directed $s \to t$ path of D.*

Other versions:

Theorem 87 (Menger (undirected, edge-version)). *Let G be a finite undirected graph and $s \neq t \in V$. There are k pairwise edge-disjoint st paths in the graph if and only if for every $s\bar{t}$ set S, $d(S) \geq k$.*

The vertex-connectivity statement of Menger's theorem is as follows:

Theorem 88 (Menger (directed, vertex-version)). *Let D be a finite directed graph and s and t two vertices such that there is no st arc. Then the size of the minimum vertex cut for s and t (the minimum number of vertices, distinct from s and t, whose removal destroys all directed $s \to t$ paths) is equal to the maximum number of pairwise internally vertex-disjoint directed paths from s to t.*

Proof. If there are k pairwise internally vertex-disjoint directed paths from s to t, it is easy to see that we need to remove at least k vertices to destroy them.

Now suppose that the size of the minimum vertex cut is k. Create a new digraph $D' = (V', A')$. For every u vertex in D, let u' and u'' be vertices of D'. Remove s' and t''. For every $u \neq s, t$, let $u'u'' \in A'$ and for every $uv \in A$, include k+1 parallel edges of $u''v'$ in D'.

If there are k edge disjoint $s' \to t''$ paths in D' then they correspont to k internally vertex-disjoint directed paths in D. If there are less than k edge disjoint $s' \to t''$ paths in D', then from directed edge verson of the Menger theorem, there are $k-1$ edges that cover all the $s' \to t''$ paths. Because of the costruction, all this edges are of the type $u'u''$, thus they correspont to $k-1$ vertices in $V - \{s, t\}$ that cover all the $s \to t$ paths in D. This contradicts our assumtion.

$\square$

Theorem 89 (Menger (undirected, vertex-version)). *Let G be a finite undirected graph and s and t two nonadjacent vertices. Then the size of the minimum vertex cut for s and t (the minimum number of vertices, distinct from s and t, whose removal disconnects s and t) is equal to the maximum number of pairwise internally vertex-disjoint paths from s to t.*

Lecture 23 (June 28) Flows

We have a direcred graph $D = (V, E)$, $s, t \in V$ are special vertices (source and target) and there is a non-negative function $c : E \to \mathbb{R}_+$, called the capacity. We call $N := (D, s, t, c)$ a network.

A function $f : E \to \mathbb{R}_+$ is a (feasible) flow, if it satisfies:

- $0 \leq f(ij) \leq c(ij)$ for every $ij \in E$
- Kirchhoff's law: $\sum_{j:ji \in E} f(ji) = \sum_{j:ij \in E} f(ij)$ for every vertex $i \in V$ sucht that $i \neq s$ and $i \neq t$.

Kirchhoff's law means flow preservation: in the intermediate vertices, what flows in , also flows out.

Definition 37. The value of a flow is $v(f) = \sum_{j:sj\in E} f(sj) - \sum_{js:\in E} f(sj)$.

This is the amount of flow leaving s minus the amount of flow entering s (Sometimes we assume that there is no edge entering s, then we can omit the second part. If we want to be more general, keep it.)

Lemma 90. $v(f) = \sum_{j:jt\in E} f(jt) - \sum_{tj:\in E} f(tj)$

This Lemma claims that the amount of flow leaving s will enter t in the end.

The capacity of the cut defined by A, B, where $A, B \subseteq V$, $A \cap B = \emptyset$, $A \cup B = V$ $s \in A$ and $t \in B$ is

$$c(A,B) = \sum_{ij\in E, i\in A, j\in B} c(ij)$$

Theorem 91 (Max Flow Min Cut, Ford- Fulkerson). *In every network, the maximum total value of a flow equals the minimum min-cut capacity of a cut.*

Idea of proof: First, we can see that for every flow and any cut $v(f) \leq c(A, B)$

$$v(f) = \sum_{j:sj\in E} f(sj) - \sum_{js:\in E} f(sj) = \sum_{j:sj\in E} f(sj) - \sum_{js:\in E} f(sj) + \sum_{i\in A, i\neq s} \left(\sum_{j:ij\in E} f(ij) - \sum_{ji:\in E} f(ji) \right)$$

$$= \sum_{ij\in E, i\in A, j\in B} f(ij) - \sum_{ji\in E, i\in A, j\in B} f(ij) \leq \sum_{ij\in E, i\in A, j\in B} c(ij)$$

We use that $f(ij) \leq c(ij)$ for every edge leaving A, and that $f(ji) \geq 0$ for every edge entering A.
We have equality if and only if $f(ij) = c(ij)$ for every edge leaving A, and that $f(ji) = 0$ for every edge entering A.

We start from $f \equiv 0$ and use the Ford- Fulkerson algorithm to find augmenting paths, and in the end reach an optimal flow.

Theorem 92 (Edmonds- Karp). *Fi in the Ford Fulkerson algorihms, we always pick the shortest augmenitng path, then the alg. terminiates after $O(|V||E|)$ augmenting steps.*

One increasing step among a path takes $O(|E|)$ time, so the total running time of the algorithm is $O(|V||E|^2)$

Lecture 24-25 (July 2,5)

Proving Menger's theorem (directed edge version) form the Max flow Min cut theorem.

Flows with vertex capacities.

Generating functions

Definition 38. Let $(a_0, a_1, a_2, \ldots)$ be a sequence of real numbers. By the generating function of this sequence we understand the power series $a(x) = a_0 + a_1 x + a_2 x^2 + \ldots$

Theorem 93. *(Generalized binomial theorem). For an arbitrary real number r and for any nonnegative integer k, we define the binomial coefficient $\binom{r}{k}$ by the formula k*

$$\binom{r}{k} = \frac{r(r-1)(r-2)...(r-k+1)}{k!}$$

(in particular, we set $\binom{r}{0} = 1$). Then the function $(1+x)^r$ is the generating function of the sequence $(\binom{r}{0}, \binom{r}{1}, \binom{r}{2}\binom{r}{3}\dots)$ The power series $\binom{r}{0} + \binom{r}{1}x + \binom{r}{2}x^2 \dots$ always converges for all $|x| < 1$.

A proof belongs to the realm of calculus, and it is can be done via the Taylor series. For combinatorial applications, it is important to note that for r being a negative integer, the binomial coefficient $\binom{r}{k}$ can be expressed using the "usual" binomial coefficient (involving nonnegative integers only):
$\binom{r}{k} = (-1)^k\binom{-r+k-1}{k} = (-1)^k\binom{-r+k-1}{-r-1}$ Hence for negative integer powers of $1-x$ we obtain

$$\frac{1}{(1-x)^n} = \binom{n-1}{n-1} + \binom{n}{n-1}x + \binom{n+1}{n-1}x^2 + \cdots \binom{n+k-1}{n-1}x^k + \cdots$$

Note that the equality $\frac{1}{1-x} = 1 + x + x^2 + x^3 + \cdots$ is a particular case for $n = 1$.

Problem 18. *A box contains 30 red, 40 blue, and 50 white balls; balls of the same color are indistinguishable. How many ways are there of selecting a collection of 70 balls from the box?*

Problem 19. *A coffee shop sells three kinds of cakes - Danish cheese cakes, German chocolate cakes, and brownies. How many ways are there to buy 12 cakes in such a way that at least 2 cakes of each kind are included, but no more than 3 German chocolate cakes? Express the required number as a coefficient of a suitable power of x in a suitable product of polynomials. Solve the problem with the help of generating functions.*

Problem 20. *a) Show that the set $\{1, 2, \dots n\}$ can be partitioned into two non-empty sets in precisely $2^{n-1} - 1$ ways.*
b) Let s_n denote the number of ways the set $\{1, 2, \dots n\}$ can be partitioned into three non-empty sets. for example $s_4 = 6$.
Show that s_n is determined by the recurrence relation $s_0 = 0$, $s_1 = 0$ and $s_n = 3s_{n-1} + 2^{n-2} - 1$ if $n > 1$.
Use a generating function to find an explicit formula for s_n.

Solution:
a) Take all the non-empty subsets of the set $\{1, 2, \dots n\}$, then divide it by two.
b) If the set $\{n\}$ is one of the three sets, see part a) to to find the other two sets. If the set $\{n\}$ is not one of the three sets, there are s_{n-1} ways to partition $\{1, 2, \dots n-1\}$ to three sets, and then we can add n to any of them. Thus $s_n = 3s_{n-1} + 2^{n-2} - 1$ if $n > 1$
Let $s(x) = s_0 + s_1x + s_2x^2 + \cdots$
Let $3s(x) = 3s_0 + 3s_1x + 3s_2x^2 + \cdots$
From the above recurrence relation

$$(1-3x)s(x) = (2^0-1)x^2 + (2^1-1)x^3 + (2^2-1)x^4 + \cdots = x^2\left(\frac{1}{1-2x} - \frac{1}{1-x}\right)$$

$$s(x) = x^2 \left(\frac{1}{(1-2x)(1-3x)} - \frac{1}{(1-x)(1-3x)} \right)$$

$$s(x) = x^2 \left(\frac{3}{2(1-3x)} + \frac{1}{2(1-x)} - \frac{2}{1-2x} \right)$$

We can deduce that

$$s_n = \frac{3}{2}3^{n-2} + \frac{1}{2} - 2 \cdot 2^{n-2} = \frac{1}{2}(3^{n-1} + \frac{1}{2} - 2^n)$$

Binary trees

We are going to consider the so-called binary trees, which are often used in data structures. For our purposes, a binary tree can concisely be defined as follows: a binary tree either is empty (it has no vertex), or consists of one distinguished vertex called the root, plus an ordered pair of binary trees called the left subtree and right subtree.

Let b_n denote the number of binary trees with n vertices. Our goal is to find a formula for b_n. By listing all small binary trees, we findthat $b_0 = 1, b_1 = 1, b_2 = 2$,and $b_3 = 5$. This can serve for checking the result obtained below.

As usual, we let $b(x) = b_0 + b_1x + b_2x^2 + \cdots$ be the corresponding generating function.

$b_n = b_0b_{n-1} + +b_1b_{n-2} + \cdots + b_{n-1}b_0$.

$b(x) = 1 + xb(x)^2$.

$$b(x) = \frac{1+\sqrt{1-4x}}{2x} \quad \text{or} \quad \frac{1-\sqrt{1-4x}}{2x}$$

This seems as if there are two possible solutions. But we know that the sequence $(b_0, b_1, b_2, \dots)$, and thus also its generating function, are determined uniquely. Since $b(x)$ is continuous as a function of x (whenever it converges), we must take either the first solution (the one with "+") for all x, or the second solution (the one with "−") for all x. If we look at the first solution, we find that for x tending to 0 it goes to ∞, while the generating function $b(x)$ must approach $b_0 = 1$. So whenever $b(x)$ converges, it must converge to the second solution $b(x) = \frac{1-\sqrt{1-4x}}{2x}$.

It remains to calculate the coefficients of this generating function. To this end, we make use of the generalized binomial theorem.

$$\sqrt{1-4x} = \sum_{k=0}^{\infty} (-4)^k \binom{1/2}{k} x^k$$

$$b_n = -\frac{1}{2}(-4)^{n+1} \binom{1/2}{n+1}$$

By further manipulations, one can obtain a nicer form:

$$b_n = \frac{1}{n+1}\binom{2n}{n}$$

So the number of binary trees is a Catalan number.

Lecture 27 (July 12)

Prove Dilworth's theorem from Kőnig's theorem

Reminder:

Theorem 94 (Kőnig)**.** *In a bipartite graph G, $\nu(G) = \tau(G)$, where $\nu(G)$ is the size of the maximum matching and $\tau(G)$ is the size of the minimum vertex cover.*

Theorem 95 (Dilworth)**.** *In a partially ordered set P, the size of the maximum antichain equals the minimum number k of chains such that P can be partitioned into k chains.*

The elements of poset P are $\{p_1, p_2 \dots p_n\}$. Create a bitarite graph $G = (X, Y; E)$ $X = \{x_1, x_2 \dots x_n\}$, $Y = \{y_1, y_2 \dots y_n\}$. Connect x_i and y_j with an edge it and only if $p_i > p_j$. (Do not connect x_i and y_i.)

Lemma 96. *If M is a matching in G, we can find a partition of poset P to $n - |M|$ chains.*

Proof. Let x_i be a node that is not covered my M. For every such x_i node, we construct a C_i chain. If y_i is not covered by M, the chain is simply $\{p_i\}$.

If y_i is covered by M, suppose its pair in M is x_{i_2}, then look at y_{i_2}, its pair in M is x_{i_3} ... continue until we reach an y_{i_k} that is not covered by M.
This way $p_i < p_{i_2} < p_{i_3} < \cdots < p_{i_k}$, so it is a chain. The chains we construct this way are disjoint and cover P, so we have partition of poset P to $n - |M|$ chains. □

Lemma 97. *Let $L \subseteq X \cup Y$ a minimum vertex cover in G. Then P has an antichain of size $n - |L|$.*

Proof. If M is a maximum matching and L is a minimum vertex cover, L contains exactly one endpoint of each edge in M.

First, we show that x_i and y_i cannot both be in L. Suppose for contradiction that $x_i \in L$, $y_i \in L$. Let the partner of x_i in M be y_j, and the partner of y_i in M be x_k. Then $p_k > p_i > p_j$ so $p_k p_j$. Therefore $x_k y_j$ is an edge of the graph, but it is not covered by L.

Now, let $A = \{p_i : x_i \notin L, y_i \notin L\}$. This is an antichain in P.

□

If poset P has a antichain A, we need at least $|A|$ chains to cover the poset. From Kőnig's theorem, there is a matching M and vertex cover L such that $|M| = |L| = \nu$. Using the previous two lemmata, we can find partition of poset P to $n - \nu$ chains and an antichain of size $n - \nu$. This proves Dilworth's theorem.