Kattis

Combinatorics

Atli FF

October 27, 2024

School of Computer Science
Reykjavík University

Today's material

- Counting
- Combinatorics
- Game theory
- Probability

Combinatorics

Combinatorics

Combinatorics is study of countable discrete structures.

Combinatorics

Combinatorics is study of countable discrete structures.

Generic enumeration problem: We are given an infinite sequence of sets $A_1, A_2, \ldots A_n, \ldots$ which contain objects satisfying a set of properties. Determine

$$a_n := |A_n|$$

for general n.

The two rules

- If you have n options, you can pick one in n ways clearly.
- If you can choose one of n options **or** one of m options then there are $n + m$ ways to do so.
- If you can choose one of n options **and** one of m options then there are nm ways to do so.
- Most things can be derived from these two rules really.

Examples

- Factorial

$$n! = 1 \cdot 2 \cdot 3 \cdots n$$

- Binomial coefficient

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

Examples

- Factorial

$$n! = 1 \cdot 2 \cdot 3 \cdots n$$

- Binomial coefficient

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

  Number of ways to choose k objects from a set of n objects, ignoring order.

Binomial properties

Properties

- $$\binom{n}{k} = \binom{n}{n-k}$$
- $$\binom{n}{0} = \binom{n}{n} = 1$$
- $$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Pascal triangle!

$$
\begin{array}{ccccccccccccccccccccc}
 & & & & & & & & & & \binom{0}{0} & & & & & & & & & & \\
 & & & & & & & & & \binom{1}{0} & & \binom{1}{1} & & & & & & & & \\
 & & & & & & & & \binom{2}{0} & & \binom{2}{1} & & \binom{2}{2} & & & & & & \\
 & & & & & & & \binom{3}{0} & & \binom{3}{1} & & \binom{3}{2} & & \binom{3}{3} & & & & & \\
 & & & & & & \binom{4}{0} & & \binom{4}{1} & & \binom{4}{2} & & \binom{4}{3} & & \binom{4}{4} & & & & \\
 & & & & & \binom{5}{0} & & \binom{5}{1} & & \binom{5}{2} & & \binom{5}{3} & & \binom{5}{4} & & \binom{5}{5} & & & \\
 & & & & \binom{6}{0} & & \binom{6}{1} & & \binom{6}{2} & & \binom{6}{3} & & \binom{6}{4} & & \binom{6}{5} & & \binom{6}{6} & & \\
 & & & \binom{7}{0} & & \binom{7}{1} & & \binom{7}{2} & & \binom{7}{3} & & \binom{7}{4} & & \binom{7}{5} & & \binom{7}{6} & & \binom{7}{7} & \\
 & & \binom{8}{0} & & \binom{8}{1} & & \binom{8}{2} & & \binom{8}{3} & & \binom{8}{4} & & \binom{8}{5} & & \binom{8}{6} & & \binom{8}{7} & & \binom{8}{8} \\
 & \binom{9}{0} & & \binom{9}{1} & & \binom{9}{2} & & \binom{9}{3} & & \binom{9}{4} & & \binom{9}{5} & & \binom{9}{6} & & \binom{9}{7} & & \binom{9}{8} & & \binom{9}{9} \\
\binom{10}{0} & & \binom{10}{1} & & \binom{10}{2} & & \binom{10}{3} & & \binom{10}{4} & & \binom{10}{5} & & \binom{10}{6} & & \binom{10}{7} & & \binom{10}{8} & & \binom{10}{9} & & \binom{10}{10}
\end{array}
$$

Other useful identities

- $$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$
- $$\sum_{k=0}^{n} \binom{n}{k}^2 = \binom{2n}{n}$$
- The "hockey stick sum"

$$\sum_{k=0}^{m} \binom{n+k}{k} = \binom{n+m+1}{m}$$

Example

How many rectangles can be formed on a $m \times n$ grid?

Example

How many rectangles can be formed on a $m \times n$ grid?

The rectangle is defined by covering some x-interval and some y-interval, we can choose these independently. If we have values $1, \ldots, k$ and want to choose an interval, how can we do this?

Example

How many rectangles can be formed on a $m \times n$ grid?

The rectangle is defined by covering some x-interval and some y-interval, we can choose these independently. If we have values $1, \ldots, k$ and want to choose an interval, how can we do this?

Well we want to choose 2 out of k values, without ordering since that's chosen by the values of the numbers. Thus this gives $\binom{k}{2}$ choices. Final answer is thus $\binom{n}{2}\binom{m}{2}$.

Multinomial

What if we have many objects with the same value?

Multinomial

What if we have many objects with the same value?

- Number of permutations on n objects, where n_i is the number of objects with the i-th value.(Multinomial)

$$\binom{n}{n_1, n_2, \ldots, n_k} = \frac{n!}{n_1! n_2! \cdots n_k!}$$

Multinomial

What if we have many objects with the same value?

- Number of permutations on n objects, where n_i is the number of objects with the i-th value.(Multinomial)

$$\binom{n}{n_1, n_2, \ldots, n_k} = \frac{n!}{n_1! n_2! \cdots n_k!}$$

- Number of way to choose k objects from a set of n objects with, where each value can be chosen more than once.

$$\binom{n+k-1}{k} = \frac{(n+k-1)!}{k!(n-1)!}$$

Balls and boxes

How many different ways can we divide k identical balls into n boxes?

Balls and boxes

How many different ways can we divide k identical balls into n boxes?

- Same as number of nonnegative solutions to

$$x_1 + x_2 + \ldots + x_n = k$$

Balls and boxes

How many different ways can we divide k identical balls into n boxes?

- Same as number of nonnegative solutions to

$$x_1 + x_2 + \ldots + x_n = k$$

- Let's imagine we have a bit string consisting only of 1 of length $n + k - 1$

$$\underbrace{1\,1\,1\,1\,1\,1\,1\ldots1}_{n+k-1}$$

Stars and bars

- Choose $n-1$ bits to be swapped for 0

$$1 \ldots 1 0 1 \ldots 1 0 \ldots 0 1 \ldots 1$$

Stars and bars

- Choose $n-1$ bits to be swapped for 0

$$\underbrace{1\ldots1}_{x_1}0\underbrace{1\ldots1}_{x_2}0\ldots0\underbrace{1\ldots1}_{x_n}$$

- Then total number of 1 will be k, each 1 representing an each element, and separated into n groups

Stars and bars

- Choose $n-1$ bits to be swapped for 0

$$\underbrace{1\ldots1}_{x_1}0\underbrace{1\ldots1}_{x_2}0\ldots0\underbrace{1\ldots1}_{x_n}$$

- Then total number of 1 will be k, each 1 representing an each element, and separated into n groups
- Number of ways to choose the bits to swap

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

Example

How many different lattice paths are there from $(0, 0)$ to (n, m)?

Example

How many different lattice paths are there from $(0, 0)$ to (n, m)?

- There is 1 path to $(0, 0)$

Example

How many different lattice paths are there from $(0, 0)$ to (n, m)?

- There is 1 path to $(0, 0)$
- There is 1 path to $(1, 0)$ and $(0, 1)$

Example

How many different lattice paths are there from $(0, 0)$ to (n, m)?

- There is 1 path to $(0, 0)$
- There is 1 path to $(1, 0)$ and $(0, 1)$
- Paths to $(1, 1)$ is the sum of number of paths to $(0, 1)$ and $(1, 0)$.

Example

How many different lattice paths are there from $(0, 0)$ to (n, m)?

- There is 1 path to $(0, 0)$
- There is 1 path to $(1, 0)$ and $(0, 1)$
- Paths to $(1, 1)$ is the sum of number of paths to $(0, 1)$ and $(1, 0)$.
- Number of paths to (i, j) is the sum of the number of paths to $(i - 1, j)$ and $(i, j - 1)$.

Example

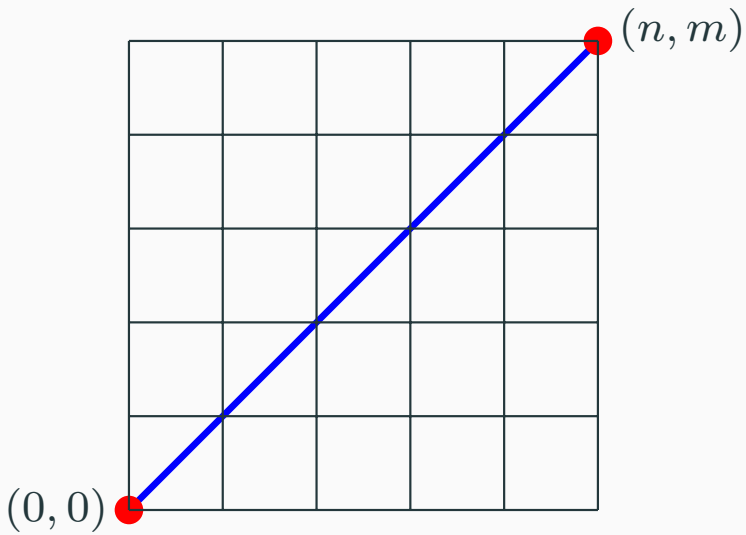
How many different lattice paths are there from $(0, 0)$ to (n, m)?

- There is 1 path to $(0, 0)$
- There is 1 path to $(1, 0)$ and $(0, 1)$
- Paths to $(1, 1)$ is the sum of number of paths to $(0, 1)$ and $(1, 0)$.
- Number of paths to (i, j) is

$$\binom{i+j}{i}$$

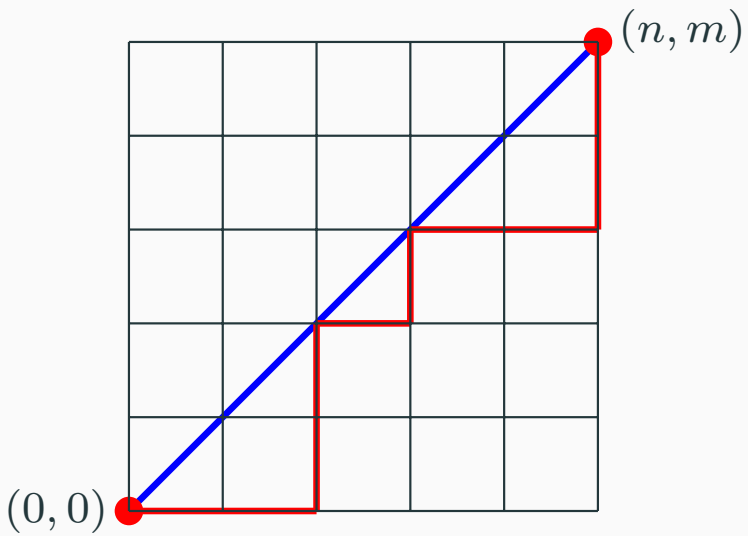
Dyck

What if we are not allowed to cross the main diagonal?



Dyck

What if we are not allowed to cross the main diagonal?

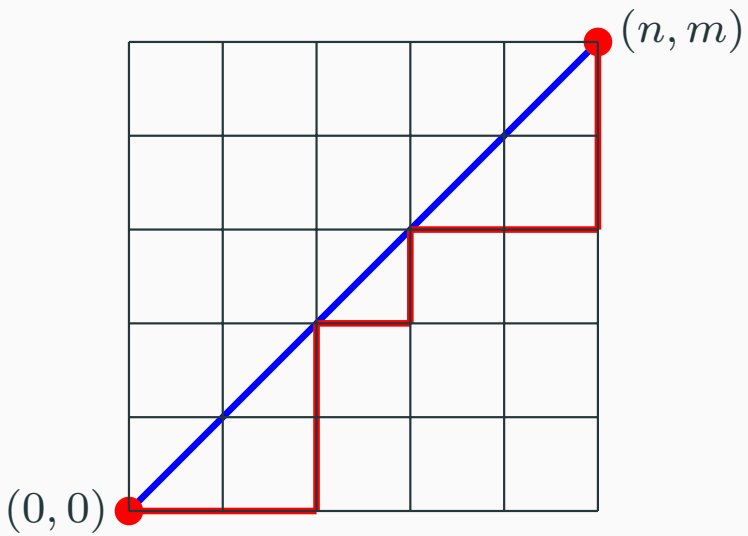


- The number of paths from $(0, 0)$ to (n, m)

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

Dyck

What if we are not allowed to cross the main diagonal?



- The number of paths from $(0, 0)$ to (n, m)

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

- C_n are known as Catalan numbers.
- Many problems involve solutions given by the Catalan numbers.

More Catalan

- Number of different ways $n+1$ factors can be completely parenthesized.

$$((ab)c)d \quad (a(bc))d \quad (ab)(cd) \quad a((bc)d) \quad a(b(cd))$$

More Catalan

- Number of different ways $n+1$ factors can be completely parenthesized.

$$((ab)c)d \quad (a(bc))d \quad (ab)(cd) \quad a((bc)d) \quad a(b(cd))$$

- Number of stack sortable permutations of length n.

More Catalan

- Number of different ways $n+1$ factors can be completely parenthesized.

$$((ab)c)d \quad (a(bc))d \quad (ab)(cd) \quad a((bc)d) \quad a(b(cd))$$

- Number of stack sortable permutations of length n.
- Number of different triangulations convex polygon with $n+2$ sides

More Catalan

- Number of different ways $n+1$ factors can be completely parenthesized.

$$((ab)c)d \quad (a(bc))d \quad (ab)(cd) \quad a((bc)d) \quad a(b(cd))$$

- Number of stack sortable permutations of length n.
- Number of different triangulations convex polygon with $n+2$ sides
- Number of full binary trees with $n+1$ leaves.

More Catalan

- Number of different ways $n+1$ factors can be completely parenthesized.

$$((ab)c)d \quad (a(bc))d \quad (ab)(cd) \quad a((bc)d) \quad a(b(cd))$$

- Number of stack sortable permutations of length n.
- Number of different triangulations convex polygon with $n+2$ sides
- Number of full binary trees with $n+1$ leaves.
- And a lot more.

Examples

Working up to π

- If we have n options, but m of them are forbidden, we are left with $n - m$ ways to choose.
- This means if we have n options and m options, but k of them are in common between them, that's $n + m - k$ different options. This can be rewritten as $|A \cup B| = |A| + |B| - |A \cap B|$.
- We can keep going with this for larger numbers of sets, say options from A, B and C. Then to know our total number of options we have to subtract what is in common. But if we do that thrice we will have removed what is in common between all three, which we need to add back.
- In total this becomes

$$|A \cup B \cup C| = |A|+|B|+|C|-|A \cap B|-|A \cap C|-|B \cap C|+|A \cap B \cap C|$$

PIE

- By generalizing this to n sets using induction you can get an equation for the size of the union of n sets $A_1, \ldots, A_n$. This equation is usually called the principle of inclusion-exclusion, or PIE.

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\substack{J \subseteq \{1,\ldots,n\} \\ J \neq \varnothing}} (-1)^{|J|-1} \left|\bigcap_{j \in J} A_j\right|$$

Permutations

- Let's look at permutations. Often it's convenient to denote them by a **bijective** function $\pi : [n] \to [n]$ where $[n] \coloneqq \{1, \ldots, n\}$.
- As mentioned before there are $n!$ permutations of $[n]$, but there's much more to it. A *fixed point* of a permutation π is an x such that $\pi(x) = x$. Things we could consider regarding this include: how many permutations have no fixed points? Such permutation sare called derangements.
- To count the number of derangements, we'll use a classic trick called counting the complement.
- We know how many permutations there are, so we'll count how many permutations have *some* fixed point and subtract that from the total.

Fixed points

- Let A_i denote all permutations of $[n]$ where i is a fixed point. Then we can permute everything but i as usual, so $|A_i| = (n-1)!$. But if we want several values to be fixed, say everything in J, then we can use the same argument to get that there are $(n - |J|)!$ such permutations. Thus $\bigcap_{j \in J} A_j$ has $(n - |J|)!$ elements.
- Thus we can now use PIE to count the size of $\bigcup_{j \in J} A_j$, which is what we want. We just have to keep in mind that the intersection of k different A_j will appear $\binom{n}{k}$ times. Thus

$$\sum_{i=1}^{n} (-1)^{i-1} \binom{n}{i} (n-i)! = n! \sum_{i=1}^{n} (-1)^{i-1} \frac{1}{i!}$$

Derangement

- Thus the number of derangements is just $n!$ minus this. We denote this by $!n$ and get

$$!n = n! \sum_{i=0}^{n} \frac{(-1)^i}{i!}$$

- A fun side effect of this is that $n!/!n$ very quickly approaches e as n grows.

Permutations and sign

- Permutations of $[n]$ have a whole lot more interesting combinatorial properties, but the two that appear the most in these kinds of problems are sign and inversion number.
- Note that if σ is a permutation we can look at $x, \sigma(x), \sigma(\sigma(x)), \ldots$. Since $[n]$ is finite, this will eventually loop and we get a cycle in the permutation.
- By considering what cycle each element belongs to, we can factor permutations into cycles. Thus we get a cycle form of permuations, so we can write for example $(xyz)(ab)(t)$ to denote $\sigma(x) = y$, $\sigma(y) = z$, $\sigma(z) = x$, $\sigma(a) = b$, $\sigma(b) = a$ and $\sigma(t) = t$. Note that the contents of the parentheses are disjoint.

Sign

- Furthermore we can rewrite $(abc\ldots x)$ as $(ab)(ac)\ldots(ax)$ so all permutations can be written as a sequence of swaps. It can be proved that the parity of the number of swaps is fixed for a given permutation. So we call this parity its *sign*.
- We denote this by $\operatorname{sgn}(\pi)$ and make it 1 when the number of swaps is even and -1 when the number of swaps is odd. Then we get that $\operatorname{sgn}(\pi_1 \circ \pi_2) = \operatorname{sgn}(\pi_1)\operatorname{sgn}(\pi_2)$ and $\operatorname{sgn}(\mathrm{id}) = 1$. This also means that $\operatorname{sgn}(\pi^{-1}) = \operatorname{sgn}(\pi)$.

Inversions

- Take $i, j \in [n]$. The ordered pair (i, j) is called an *inversion* of π if $i < j$ but $\pi(i) > \pi(j)$.
- The inversion number of π is then the number of such swaps.
- But how can this be calculated efficiently?
- One is to use segment trees, but another is to use mergesort and count how many inversions get fixed as you go.

Invnum - Mergesort version

```
ll merge(vi& v, vi& l, vi& r) {
  ll i = 0, j = 0, cnt = 0;
  while(i < l.size() || j < r.size()) {
    if(i == l.size()) v[i + j] = r[j], ++j;
    else if(j == r.size()) v[i + j] = l[i], ++i;
    else if(l[i] <= r[j]) v[i + j] = l[i], ++i;
    else v[i + j] = r[j], cnt += l.size() - i, ++j;
  } return cnt; }

ll invnum(vi &v) { if(v.size() < 2) return 0;
  int m = v.size() / 2; vi l(m), r(v.size() - m);
  copy(v.begin(), v.begin() + m, l.begin());
  copy(v.begin() + m, v.end(), r.begin());
  return invnum(l) + invnum(r) + merge(v, l, r); }
```

A counting problem

- Let us consider an example from Kattis of a harder counting problem. We get $1 \leq n \leq 1000$ and $1 \leq c \leq 10000$ and want to find the number of permutations of n elements with c inversions. The answer should be returned modulo $10^9 + 7$.
- Where do we start?
- Can we solve the problem for n, c in terms of smaller values?

Subdivision

- Denote the answer for n, c by $f(n, c)$ and let us write our permutation as a list $\pi(1), \ldots, \pi(n)$. We can make a permutation of $n+1$ elements by adding $n+1$ to this list. If we add it at location k (and shift what's right of it) we add $n-k+1$ inversions.
- We can also do this backwards, starting with a permutation on n elements and removing n. Let us consider permutations on n lements with c inversions and the value n is t-th last in the list.
- Then if we remove n we fix $t-1$ inversions, so we end up with a permutation on $n-1$ elements with $c-t+1$ inversions. Thus for this particular t we get $f(n-1, c-t+1)$. By considering all t we get:

$$f(n, c) = \sum_{i=0}^{\min(c, n-1)} f(n-1, c-i)$$

Dynamic programming

- Now we have a recursive formula in terms of (n, c), so we can use dynamic programming! But the time complexity is $\mathcal{O}(nc^2)$ because there are nc states and each takes c time to calculate.
- We use the same trick as two days ago, we precalculate the sum so each value will only take constant time. For this we need bottom-up dynamic programming. So we define $p(n, c)$ as

$$p(n, c) = \sum_{i=0}^{c} f(n, i)$$

- Then we can update both p and f in constant time and solve the problem in $\mathcal{O}(nc)$ which is good enough!

The code

```cpp
#include <bits/stdc++.h>
using namespace std;
const int mod = 1e9 + 7;

int n, c;
int dp[1005][10005];
int pr[1005][10005];

int main() {
    cin >> n >> c;
    if(c == 0) {
        cout << "1\n";
        return 0;
    }
    for(int i = 0; i <= c; ++i) dp[0][i] = 0;
    for(int i = 0; i < n; ++i) dp[i][0] = 1;
    for(int i = 1; i < n; ++i) {
        pr[i - 1][0] = dp[i - 1][0];
        for(int j = 1; j <= c; ++j) {
            pr[i - 1][j] = (pr[i - 1][j - 1] + dp[i - 1][j]) % mod;
        }
        for(int j = 1; j <= c; ++j) {
            dp[i][j] = pr[i - 1][j];
            if(j >= i + 1) {
                dp[i][j] -= pr[i - 1][j - i - 1];
                dp[i][j] = (dp[i][j] % mod + mod) % mod;
            }
        }
    }
    cout << dp[n - 1][c] << '\n'; }
```

Combinatorial equations

Linear recurrences

- Let us define a linear recurrence. We say a sequence of numbers follows a linear recurrence if they are defined using some initial values and then the subsequent values being defined by a linear equation of the form

$$a_n = \lambda_1 a_{n-1} + \lambda_2 a_{n-2} + \cdots + \lambda_k a_{n-k} + \lambda$$

  with the λ_is constant. k is said to be the degree of the recurrence relation.

- An example of this is the fibonacci numbers, where $\lambda_1 = \lambda_2 = 1$, $\lambda = 0$ and $a_1 = a_2 = 1$.

Calculating a sequence

- The trick is that using matrices we can calculate values in these sequences very fast. We can get the n-th number in $\mathcal{O}(k^3 \log(n))$ time. Since k is usually very small, almost always < 10, this is quite fast. For example for fibonacci we have $k = 2$. The trick is the following equation:

$$\begin{pmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_{k-1} & \lambda_k & \lambda \\ 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}^n \begin{pmatrix} a_k \\ a_{k-1} \\ \vdots \\ a_2 \\ a_1 \\ 1 \end{pmatrix} = \begin{pmatrix} a_{n+k} \\ a_{n+k-1} \\ \vdots \\ a_{n+2} \\ a_{n+1} \\ 1 \end{pmatrix}$$

Linear recurrence

- So for example we can calculate the fibonacci numbers using:

$$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

- We omitted one row there since $\lambda = 0$. But similarly if $a_1 = 1, a_2 = 2$ and $a_n = 3a_{n-1} - a_{n-2} + 6$ the corresponding calculation would be

$$\begin{pmatrix} 3 & -1 & 6 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}^n \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

Probability

Probability theory

- We will now briefly talk about probabilities.
- Some problems use probabilities, but only really on a surface level so we won't delve deep.
- Sometimes there are slightly more involved problems about markov chains and the like, but we won't go much into that here.
- We won't formally define probabilities, measures, distributions and the like here. If you want to learn that take a probabilistics course.

Discrete basics

- In these problems we consider only discrete probabilities.
- Then there are finitely (or at least countably) many options that can happen, $x_1, \ldots, x_n$, and each of those has some probability $0 \leq P(x_i) \leq 1$ of occurring, such that $\sum P(x_i) = 1$. An **event** X is then a set of such outcomes x_i.
- If the x_i correspond to numbers we can speak of expected values. $E[X]$ and is calculated using $E[X] = \sum x_i P(x_i)$.
- The most important property of expected values is that it's linear. That is to say if we have some distributions X and Y and a is some number, then $E[X + Y] = E[X] + E[Y]$ and $E[aX] = aE[X]$.

Game theory

Game theory

- Game theory is the theory on games. The games we will consider are between players who always play perfectly and there is no hidden information (cards on hand or something).
- The games are defined by states and what states can be reached from each of them. This means that for each state there is some set of legal moves. Some states will also be losing states, for tic-tac-toe this would for example be states where the opponent has three in a row.
- For these kinds of games that only allow a finite number of moves we can work backwards to find out whether states are going to lead to a loss or to a win.

Working backwards

- We start by collecting all end states (i.e. states where the game is over, either due to a tie or one player winning).
- Then we recursively consider states. For all reachable states you pick the best one. If one of those is a losing position for the opponent then you choose that and your current state is a winning one. If none of them are losing states for the opponent you try to pick one that forces a tie. If neither are possible the current state is a losing one.
- Let us consider an example.

Simple game

- Let us consider a game where we start with n stones in a pile. Two players take turns removing either one or two stones from the pile. The player taking the last stone wins. For what n will A win and for what n will B win if they play perfectly?
- Well, if there are no stones left and it's your turn, you lose. Thus 0 is a losing state.
- If there are one or two stones left you can take the rest, so 1 and 2 are winning states.
- If there are three left, then no matter whether you take one or two stones you put the opponent in a winning state, so 3 is a losing state.

Simple game ctd.

- You can continue like this and use induction to show that n is a losing state iff $n = 0 \ (mod \ 3)$.
- Since A starts we thus get that A loses iff the starting number of stones is a multiple of three.

Harder game

- Let us consider another game with n piles, each with $k_1, \ldots, k_k$ stones. A player may now take as many stones as they wish, but only from one of the piles at a time. The one taking the last stone wins. This is a famous game theory game called nim.
- It's much harder to see a winning strategy here, but there is a trick to it. The trick is to make sure that after your move $k_1 \oplus \cdots \oplus k_n = 0$ where $\oplus$ is the XOR operation. But why might that be?
- Well, 0 is a losing state, so if $k_1 \oplus \cdots \oplus k_n = 0$ after each of your moves and $k_1 \oplus \cdots \oplus k_n \neq 0$ when it's your turn, surely you'll win. The reason we use $\oplus$ is that if this value is zero at the start of your turn, there is no way to keep it at zero.

Xorsum

- But how do we make sure it's zero after our move? We will have to assume it's not already 0.
- Let the xor-sum be X. Then we look at $X \oplus k_i$ for each i. Since $\oplus$ is associative and xor-ing twice cancels out, this will be the xor-sum of all piles but k_i. Thus there must be some pile such that $X \oplus k_j < k_j$ since $X \neq 0$. Thus we take $k_j - (X \oplus k_j) > 0$ stones from pile j, making the xor-sum 0 after our move.
- In the same vein if X starts as zero, then no matter what move we make it won't be zero afterwards. Thus a state is a losing state iff $k_1 \oplus \cdots \oplus k_n = 0$.

Grundy-Sprague

- We will take a look at a theorem we won't prove. It's called the Grundy-Sprague theorem.
- Let us have a game with no hidden information, no ties, both players make the same kinds of moves and one that is always over in a finite number of moves, ending when a player can't make any moves.
- For any such game we can define a Grundy number for every state.

Grundy-numbers

- We start by giving all ending states which lose the Grundy number 0 (for example for nim this would be when all piles are empty).
- We then define mex. It stands for minimum excluded value. It takes a set of numbers and returns the smallest non-negative integer not in the set. We then let the Grunndy number of a state be the mex of all Grundy-numbers of states we can reach.
- Then a state is a losing one iff the Grundy-number is zero. But this isn't the most useful part of the Grundy number.

Grundy-Sprague

- Imagine we have k games going on in parallel, with a player only allowed to make a move in one of them on their turn. The player loses once they can't make a move in any of them.
- Then the Grundy-Sprague theorem says that the Grundy number of a state in this game is the xor-sum of the Grundy numbers of the individual states in the games. Thus the position is a losing one iff the xor-sum of the Grundy numbers is 0.