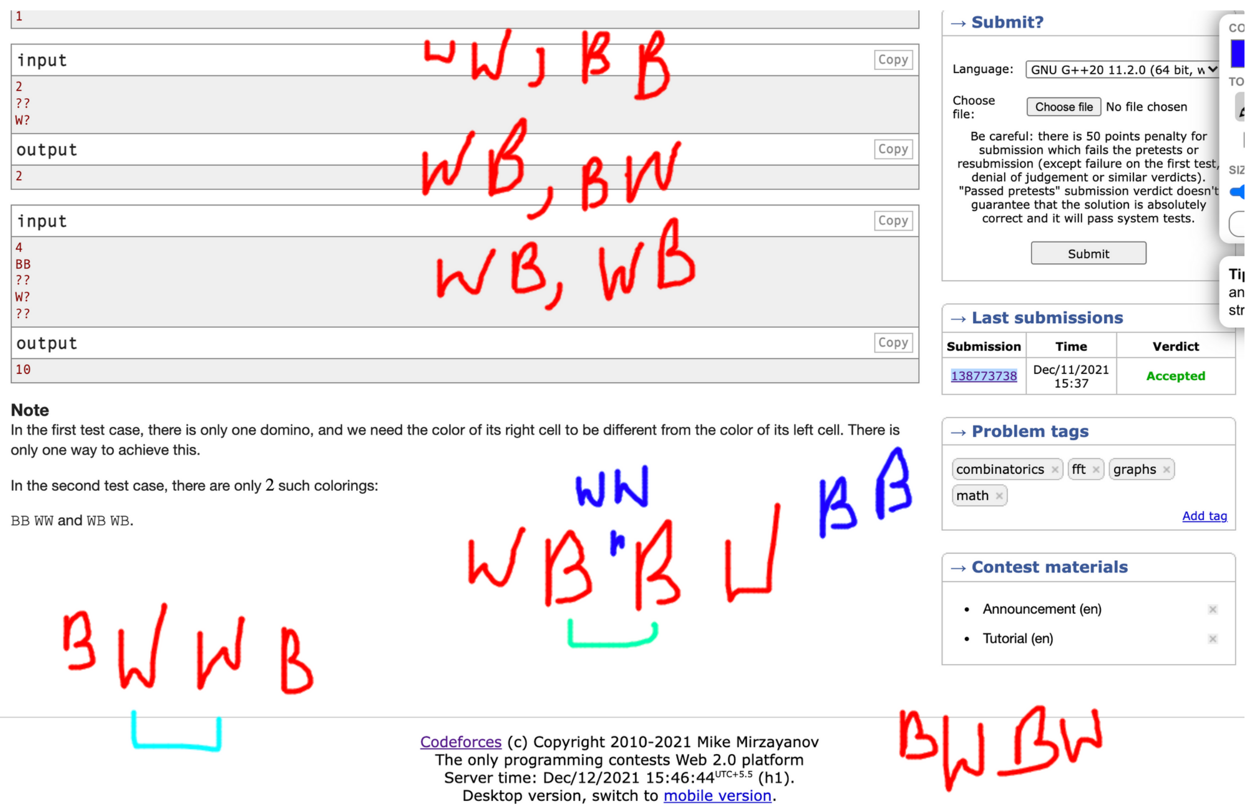
Combinatorics

Codeforces Round #758 Problem C

```
int32_t main() { IOS; int t; cin>>t; while(t--) { int n; cin>>n; vector
<pii> v1, v2; rep(i,0,n) { int a; cin>>a; v1.pb({a, i}); } rep(i,0,n) { int
b; cin>>b; v2.pb({b, i}); } sort(all(v1)); sort(all(v2)); int ans[n];
fill(ans, 0); int co[n]; fill(co, 0); set <int> s; set <int> st; for(int
i=n-1;i>=0;i--) { // cout<<v1[i].ss<<" "<<v2[i].ss<<'\n'; s.insert(v1[i].ss);
s.insert(v2[i].ss); int a=v1[i].ss, b=v2[i].ss; if(v1[i].ss!=v2[i].ss) {
if(co[a]) st.erase(a); else st.insert(a); if(co[b]) st.erase(b); else
st.insert(b); } co[v1[i].ss]++; co[v2[i].ss]++; if(st.size()==0) { for(auto
it:s) ans[it]=1; break; } } rep(i,0,n) cout<<ans[i]; cout<<"\n"; } }
```

https://codeforces.com/contest/1608/problem/C

Codeforces Round #758 Problem D


1

input
2
??
W?

output
2

input
4
BB
??
W?
??

output
10

Note

In the first test case, there is only one domino, and we need the color of its right cell to be different from the color of its left cell. There is only one way to achieve this.

In the second test case, there are only 2 such colorings:

BB WW and WB WB.

→ Submit?

Language: GNU G++20 11.2.0 (64 bit, w

Choose file: Choose file No file chosen

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

→ Last submissions

Submission	Time	Verdict
138773738	Dec/11/2021 15:37	Accepted

→ Problem tags

combinatorics × fft × graphs × math ×

Add tag

→ Contest materials

- Announcement (en)
- Tutorial (en)

Codeforces (c) Copyright 2010-2021 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Dec/12/2021 15:46:44 UTC+5.5 (h1).
Desktop version, switch to mobile version.

Factorial modulo m

$n! = \prod_{i=1}^{n} i$

```
int f[N]; f[0]=1; for(int i=1;i<=n;i++) f[i]=(f[i-1]*i)%m;
```

Combinations modulo m

$${}^{n}C_{r} = \frac{n!}{r!(n-r)!}$$

```
int f[N],invf[N]; int ncr(int n,int r) { if(n<0 || r<0 || n<r) return 0; int
ans=f[n]; ans=(ans*invf[n-r])%m; ans=(ans*invf[r])%m; return ans; } int
main() { f[0]=1; for(int i=1;i<=n;i++) f[i]=(f[i-1]*i)%m; for(int
i=0;i<=n;i++) invf[i]=inverse(f[i]); }
```

Find ${}^{n}C_{r}$ (no modulo) given $n \leq 50$

$50!$ exceeds 10^{18} but ${}^{50}C_{25}$ is less than 10^{18}

$${}^{n}C_{r} = {}^{n}C_{r-1} \times \frac{n+1-r}{r}$$

```
int ncr(int n,int r) { int ans=1; for(int i=1;i<=r;i++) { ans*=(n+1-i);
ans/=r; } return ans; }
```

You are given n indistinguishable items. Your task is to divide them into k consecutive non-empty groups. Find the number of ways to do so.

n=5, k=3

6 ways -

- 1 + 1 + 3

- 1 + 2 + 2
- 1 + 3 + 1
- 2 + 1 + 2
- 2 + 2 + 1
- 3 + 1 + 1

The number of integral solutions of $x_1 + x_2 + ... + x_k = n$ such that all $x_i \geq 1$.

= Coefficient of x^{n-k} in $(1-x)^{-k}$ = ${}^{n-1}C_{k-1}$

If empty groups are allowed, number of ways = ${}^{n+k-1}C_{k-1}$

Circular Permutations

Circular permutations with clockwise and anti-clockwise considered different = $(n-1)!$

Circular permutations with clockwise and anti-clockwise considered same = $\frac{(n-1)!}{2}$

Multinomial Theorem

You are given n items and you need to distribute them into groups of size $a_1, a_2, ..., a_k$ such that $a_1 + a_2 + ... + a_k = n$.

$$\frac{n!}{(a_1)!(a_2)!\ldots(a_k)!}$$

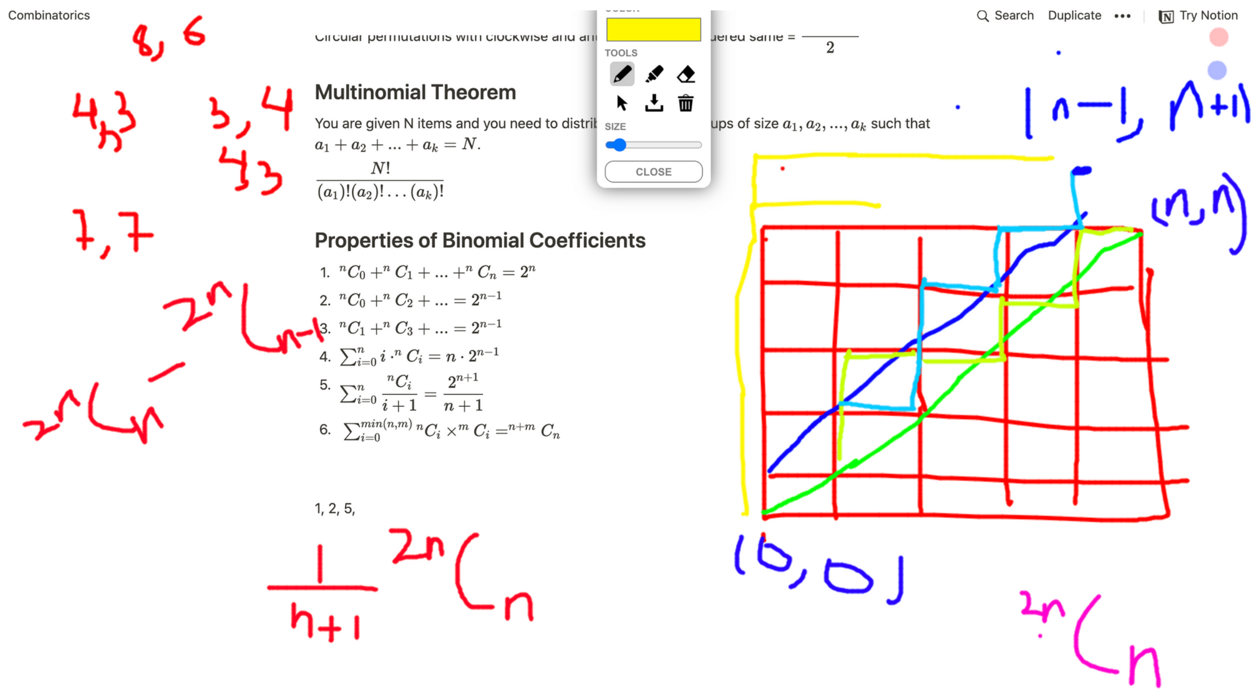
Properties of Binomial Coefficients

1. ${}^nC_0 +^n C_1 + ... +^n C_n = 2^n$
2. ${}^nC_0 +^n C_2 + ... = 2^{n-1}$
3. ${}^nC_1 +^n C_3 + ... = 2^{n-1}$
4. $\sum_{i=0}^{n} i \cdot^n C_i = n \cdot 2^{n-1}$
5. $\sum_{i=0}^{n} \frac{{}^nC_i}{i+1} = \frac{2^{n+1}}{n+1}$
6. $\sum_{i=0}^{min(n,m)} {}^nC_i \times^m C_i =^{n+m} C_n$
7. ${}^nC_r =^{n-1} C_r +^{n-1} C_{r-1}$

Catalan Numbers

1, 2, 5, 14, 42, ...

Number of ways to travel from (1, 1) to (n, n) without crossing the principal diagonal



$$C(n) = {}^{2n}C_n - {}^{2n}C_{n-1} = \frac{{}^{2n}C_n}{n+1}$$

Number of regular bracket sequences that can be formed using n pairs of brackets

n '(' n ')'

(()()), ((())) , n "+1" n "-1"

$$\frac{{}^{2n}C_n}{n+1}$$

Problem

Find the sum of XOR of all k-tuples which can be formed using elements of an array a of length n modulo $10^9 + 7$. Formally, find $\sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} (a_{i_1} \oplus a_{i_2} \oplus a_{i_3} \oplus \ldots \oplus a_{i_k}) \bmod 10^9 + 7$.
Constraints - $1 <= k <= n <= 10^5, 0 <= A_i <= 10^9$

Example - A = [1, 3, 5, 8] and k=3

k-tuples = [(1,3,5), (1,5,8), (1,3,8), (3,5,8)]

XOR of k-tuples = [7,12,10,14]

Sum = 43

Let the number of numbers in which the i-th bit is set be b_i. We can take either 1 set bit and $k-1$ unset bits or 3 set bits and $k-3$ unset bits or 5 set bits and $k-5$ unset bits and so on.

Total number of k-tuples in which i-th bit is set = $\sum_{j=1,\, j\, is\, odd}^{k} {}^{b_i}C_j \times {}^{n-b_i}\ C_{k-j}$

The i-th bit will contribute 2^i to the answer.

Hence, required answer $= \sum_{i=0}^{30}(\sum_{j=1,\, j\, is\, odd}^{k} {}^{b_i}C_j \times {}^{n-b_i}\ C_{k-j}) \times 2^i$

Time Complexity - $\mathcal{O}(n \log \max a_i)$

Homework

https://codeforces.com/contest/1204/problem/E

Inclusion-Exclusion Principle

If you are given two sets A and B then $n(A \cup B) = n(A) + n(B) - n(A \cap B)$

If you are given three sets A, B and C then $n(A \cup B \cup C) = n(A) + n(B) + n(C) - n(A \cap B) - n(A \cap C) - n(B \cap C) + n(A \cap B \cap C)$

$n(A_1 \cup A_2 \cup A_3 \cup \ldots \cup A_n) = \sum_{k=1}^{n}(-1)^{k-1}(\sum all\, k - tuples) = \sum_{k=1}^{n}(-1)^{k-1}(\sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} n(A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_k}))$

Derangements

Derangements = Number of ways to arrange n objects such that no object is present in its correct place.

Example

n=3

2, 3, 1

3, 1, 2

Suppose A_i = Set of permutation where i-th element is in its correct place.

$n!$ - Number of ways in which atleast one element is in its correct position.

$= n! - n(A_1 \cup A_2 \cup A_3 \cup \ldots \cup A_n)$

$= n! - \sum_{k=1}^{n} (-1)^{k-1} (\sum all\, k - tuples)$

$n(A_1) = (n-1)!$

$n(A_1 \cap A_2) = (n-2)!$

For any k-tuple $n(A_1 \cap A_2 \cap A_3 \cap \ldots \cap A_k) = (n-k)!$

Number of k-tuples = ${}^nC_k$

$\sum all\, k - tuples = (n-k)! \times^n C_k = \frac{n!}{k!}$

Derangements = $= n! - \sum_{k=1}^{n} (-1)^{k-1} \frac{n!}{k!}$

Problem

You are given an integer n. Find the number of integers between 1 and n which are divisible by A_1or A_2 or $\ldots$ or A_k. Constraints - All A_i are prime, $k \leq 20$ and $n \leq 10^9$.

Number of integers between 1 and n divisible by m = $\left\lfloor \frac{n}{m} \right\rfloor$.

n=6

k=2

A = [2,3]

[2,3,4,6]

Let B_i = Set of integers between 1 and n divisible by A_i.

Answer = $n(B_1 \cup B_2 \cup B_3 \cup \ldots \cup B_k) = \sum_{j=1}^{k} (-1)^{j-1} (\sum all\, j - tuples)$

```
int ans=0; for(int i=0;i<(1<<k);i++) { int p=1,c=0; for(int j=0;j<k;j++) {
if(i&(1<<j)) { p=p*a[j]; c++; if(p>n) break; } } if(c%2) ans = ans+n/p; else
ans = ans-n/p; }
```

Problem: Codeforces Placing Rooks

Let's assume all rows have at least one rook. Let's say a column has r rooks. Number of pairs of rooks attacking each other in that column = $r-1$. Let's assume there are c columns not having any rook.

1 1 1 1 1

1 2 1 1 0

1 2 2 0 0

Let's assume there are c columns not having any rook. Then there will be exactly c pairs of rooks attacking each other. We want $c=k$. Let $m=n-k$. We want to distribute n rooks in m columns such that each of these have atleast one rook.

Let A_i denote the set of placements in which the i-th of these columns is empty. Total number of placements $=m^n$. Number of placements in which each column has atleast one rook $=m^n-n(A_1 \cup A_2 \cup \ldots \cup A_m)$. Applying Inclusion-Exclusion, this is equal to

$ans_1 = m^n - {}^mC_1 \times (m-1)^n + {}^mC_2 \times (m-2)^n - \ldots = \sum_{j=0}^{m}(-1)^j \times {}^mC_j \times (m-j)^n$

Number of placements in which every row has a rook and exactly k columns are empty $= ans_2 = {}^nC_k \times ans_1$.

Number of ways in which all rows have at least 1 rook = Number of ways in which all columns have at least 1 rook (take the transpose of the board).

Total number of ways = $2 \times ans_2$

If $k=0$, subtract the number of ways in which there are exactly 1 rook in all rows as well as columns. $n!$ ways.

Lucas Theorem

Suppose we want to find ${}^nC_r \bmod m$ where $n <= 10^9, m <= 100000$

Represent n and r in base m. Then ${}^nC_r \bmod m = {}^{n_0}C_{r_0} \times {}^{n_1}C_{r_1} \times \ldots \times {}^{n_k}C_{r_k} \bmod m$ where $n = n_k n_{k-1} \ldots n_2 n_1 n_0$ in base m and $r = r_k r_{k-1} \ldots r_2 r_1 r_0$ in base m.

Application - ${}^nC_r \bmod 2 = {}^{n_0}C_{r_0} \times {}^{n_1}C_{r_1} \times \ldots \times {}^{n_k}C_{r_k} \bmod 2$.

- $n_i = 0, r_i = 0 : 1$
- $n_i = 0, r_i = 1 : 0$
- $n_i = 1, r_i = 0 : 1$

- $n_i = 1, r_i = 1 : 1$

If there is a bit which is set in r but not in n then ${}^nC_r$ is even else odd.

If r is a sub-mask of n then ${}^nC_r$ is odd else even. Condition for checking if r is a sub-mask of n - $n|r == n$ or $n\&r == r$.