

Question bank

Instructions: Write in below format

- Question
 - Constrains
 - One Example testcase
 - If possible working code, editorial
-

Question: (easy)

You are given a sequence a_1, a_2, \dots, a_n consisting of n non-zero integers (i.e. $a_i \neq 0$).

You have to calculate two following values:

- 1.the number of pairs of indices (l, r) ($l \leq r$) such that $a_l \cdot a_{l+1} \dots a_{r-1} \cdot a_r$ is negative;
- 2.the number of pairs of indices (l, r) ($l \leq r$) such that $a_l \cdot a_{l+1} \dots a_{r-1} \cdot a_r$ is positive;

Constraints

$1 \leq n \leq 2 \cdot 10^5$ (sequence length)

$-10^9 \leq a_i \leq 10^9$ (element values)

$a_i \neq 0$ (no zero elements allowed)

Memory limit: 256 megabytes per test

Testcase

5

5 -3 3 -1 1

ans 8 7

10

4 2 -4 3 1 2 -4 3 2 3

ans 28 27

```
#include <bits/stdc++.h>
using namespace std;

#define _F(i, a, b) for (ll i = a; i < b; ++i)
#define __F(i, a, b) for (ll i = a; i >= b; --i)
#define f1(a) for (ll i = 0; i < a; ++i)
#define f2(a) for (ll j = 0; j < a; ++j)
#define ll long long
#define pb push_back
#define vl vector<ll>
#define pll pair<ll, ll>
#define vll vector<pair<ll, ll>>
#define _karlofast_ ios_base::sync_with_stdio(false), cin.tie(NULL);
```

```

void solve()
{
    ll n;
    cin>>n;
    vl arr(n+1);
    arr[0] = 0;
    ll od = 0 , ev = 1, ans =0 ;
    f1(n)
    {
        cin >> arr[i+1];
        arr[i+1] = (arr[i+1] < 0);
    }
    f1(n){
        arr[i+1] += arr[i];
        if(arr[i+1]%2 == 0)
            ev++;
        else
            od++;
    }
    ans += od * ev;
    ll num = (n * (n - 1)) / 2 + n;
    cout << ans << " " << num - ans << endl;
}

int main()
{
    ll t = 1;
    // cin>>t;
    while (t--)
        solve();
}

```

Question: (med) <https://codeforces.com/problemset/problem/372/A>

binary search
greedy
sortings
two pointers
*1600

There are n envelopes with pockets. Each envelope has a size (integer number).

An envelope can go into another envelope's pocket if and only if the size of the envelope who holds the other envelope is at least twice as large as the size of the envelope who is held.

Each envelope can hold at most one envelope, and the envelope who is held by another envelope cannot hold any envelopes.

The envelope who is held by another envelope cannot be visible from outside.

Please, find a plan of holding envelopes with the minimal number of envelopes who is visible.

Constraints

$1 \leq n \leq 5 \cdot 10^5$ (number of envelopes)

$1 \leq s_i \leq 10^5$ (envelope sizes)

Testcase

8

9

1

6

2

6

5

8

3

Ans 5

8

2

5

7

6

9

8

4

2

Ans 5

```
#include<bits/stdc++.h>
```

```

using namespace std ;

//printing statements
#define trace1(x) cout <<#x<<": "<<x<<" "<<endl;
#define trace2(x, y) cout <<#x<<": "<<x<<" | "<<#y<<": "<<y<<" endl;
#define trace3(x, y, z) cout <<#x<<": "<<x<<" | "<<#y<<": "<<y<<" |
"<<#z<<": "<<z<<endl;
#define trace4(a, b, c, d) cout <<#a<<": "<<a<<" | "<<#b<<": "<<b<<"
| "<<#c<<": "<<c<<" | "<<#d<<": "<<d<<endl;
#define trace5(a, b, c, d, e) cout <<#a<<": "<<a<<" | "<<#b<<":
"<<b<<" | "<<#c<<": "<<c<<" | "<<#d<<": "<<d<<" | "<<#e<<":
"<<e<<endl;

//loops
#define _F(i,a,b) for(ll i = a ; i<b ; ++i)
#define __F(i,a,b) for(ll i = a ; i>=b ; --i)
#define f1(a) for(ll i = 0 ; i<a ; ++i)
#define f2(a) for(ll j = 0 ; j<a ; ++j)
#define f3(a) for(ll k = 0 ; k<a ; ++k)

//shorwords for long
#define ll long long
#define ld long double
#define mp make_pair
#define pb push_back
#define vl vector<ll>
#define vd vector<double>
#define pll pair<ll,ll>
#define vll vector<pair<ll,ll> >
#define vls vector<pair<ll,string> >
#define mll map<ll,ll>
#define msl map<string,ll>
#define F first
#define S second

//stl formulas
#define svl(vector) accumulate(vector.begin(), vector.end(), 0);

```

```

#define _fast_ ios_base::sync_with_stdio(false),cin.tie(NULL);
#define inf 1e18
#define MOD (1ll)(1e9 + 7)

//upper bound and lower bound
// lower bound means pehla = aur agar = nhi to pehla greater
// upper bound means last = aur agar = nhi to pehla greater
//bool cmp(pll const & a, pll const & b){return a.F != b.F ? a.F<b.F
: a.S < b.S;}

void solve(){
    ll n;
    cin>>n;
    vl arr(n);
    f1(n){
        cin>>arr[i];
    }
    sort(arr.begin(),arr.end());
    ll co = 0 ;
    ll ans = n ;
    vl::iterator indd;
    indd = arr.begin()+((n+1)/2);
    // cout<<"hi"<<endl;
    while(co<n){
        indd = lower_bound(indd,arr.end(),2*arr[co]);
        ll ind = indd - arr.begin();
        // trace2(ind, co);
        if(ind < n){
            indd++;
            ans--;
        }
        else{
            break;
        }
        co++;
    }
    cout<<ans<<endl;
}

```

```

int main()
{
    _fast_
    bool testing = false;
    if(testing)
    {
        freopen("../input.txt", "rt", stdin);
        int start = clock();
        ll t = 1;
        // cin >> t;
        while(t--){
            solve();
        }
        int end = clock();
        cout << "time: " << (end -
start)/(double)(CLOCKS_PER_SEC)*1000 << " milliseconds\n";
    }
    else
    {
        ll t = 1;
        // cin >> t;
        while(t--){
            solve();
        }
    }
}

```

Question (med)

<https://codeforces.com/problemset/problem/225/C>

dp

matrices

*1700

You've got an $n \times m$ pixel picture. Each pixel can be white or black. Your task is to change the colors of as few pixels as possible to obtain a barcode picture.

A picture is a barcode if the following conditions are fulfilled:

- All pixels in each column are of the same color.
- The width of each monochrome vertical line is at least x and at most y pixels. In other words, if we group all neighboring columns of the pixels with equal color, the size of each group cannot be less than x or greater than y .

Given the original image, print the minimum number of pixels to repaint to transform it into a barcode picture. It is guaranteed that the answer exists.

Constraints

- $1 \leq n, m, x, y \leq 1000$
- $x \leq y$
- Each of the n lines describing the original image contains exactly m characters.
- Each character is either '.' (white pixel) or '#' (black pixel).

Example Testcases

Input

...

6 5 1 2

##..#

..###

##.#.

#...#

.##.#

###..

...

Output

...

11

...

Explanation for First Testcase

After changing the minimum number of colors, the picture can look like:

...

```
##..  
##..  
##..  
##..  
##..  
##..  
...
```

Input
...

```
2 5 1 1  
#####
```

```
.....  
...
```

Output
...

```
5  
...
```

Explanation for Second Testcase

After changing the minimum number of colors, the picture can look like:

```
...  
.#.#.  
.#.#.  
...
```

Solution

```
#include <bits/stdc++.h>  
using namespace std;  
  
#define _F(i, a, b) for (ll i = a; i < b; ++i)  
#define __F(i, a, b) for (ll i = a; i >= b; --i)  
#define f1(a) for (ll i = 0; i < a; ++i)  
#define f2(a) for (ll j = 0; j < a; ++j)  
#define ll long long  
#define pb push_back
```



```

#define vl vector<ll>
#define pll pair<ll, ll>
#define vll vector<pair<ll, ll>>
#define _karlofast_ ios_base::sync_with_stdio(false), cin.tie(NULL);

ll n, m, x, y;
vl arr(1000, 0);
ll gans = INT_MAX;
ll dp[1004][1004][2];

ll go_on(ll ind, ll cont, bool val)
{
    if (ind == m)
    {
        if (cont > x)
            return 0;
        return INT_MAX;
    }
    if (dp[ind][cont][val] != -1)
        return dp[ind][cont][val];
    ll ans1 = INT_MAX, ans2 = INT_MAX;
    if (cont <= x)
    {
        ans1 = go_on(ind + 1, cont + 1, val) + (val ? arr[ind] : n - arr[ind]);
    }
    else if (cont > y)
    {
        ans2 = go_on(ind + 1, 2, not val) + (val ? n - arr[ind] : arr[ind]);
    }
    else
    {
        ans1 = go_on(ind + 1, cont + 1, val) + (val ? arr[ind] : n - arr[ind]);
        ans2 = go_on(ind + 1, 2, not val) + (val ? n - arr[ind] : arr[ind]);
    }
    return dp[ind][cont][val] = min(ans1, ans2);
}

```

```

}

void solve()
{
    cin >> n >> m >> x >> y;
    char c;
    memset(dp, -1, sizeof(dp));
    f1(n)
    {
        f2(m)
        {
            cin >> c;
            if (c == '.')
                arr[j]++;
        }
    }
    gans = min(go_on(0, 1, 0), go_on(0, 1, 1));
    cout << gans << endl;
}

int main()
{
    _karlofast_
    ll t = 1;
    // cin>>t;
    while (t--)
        solve();
}

```

Question

<https://codeforces.com/problemset/problem/453/A>
probabilities

Given a fair dice with m faces (numbered 1 through m), each face appears with probability $1/m$ on every toss. You roll the dice n times.
Calculate the expected value of the maximum number shown over all tosses.

Constraints

- $1 \leq m, n \leq 10^5$

Output precision should be 10^{-4}

Testcases

Testcase 1

6 1

3.500000000000

Testcase 2

6 3

4.958333333333

Testcase 3

2 2

1.750000000000

Testcase Explanation

For the third test case ($m = 2, n = 2$):

- Possible results when rolling two dice with faces 1 and 2:

- 1 and 2 \rightarrow maximum is 2

- 1 and 1 \rightarrow maximum is 1

- 2 and 1 \rightarrow maximum is 2

- 2 and 2 \rightarrow maximum is 2

- Probability of each outcome: 0.25

So, expected value is $\{2 + 1 + 2 + 2\}/4 = 1.75$.

Solution

```
#include<bits/stdc++.h>
using namespace std ;

#define ll long long
#define ld long double
#define trace2(x, y) cout <<#x<<": "<<x<<" | "<<#y<<":
```

```

"<y<< endl;
#define trace3(x, y, z)      cout <<#x<<": "<<x<<" | "<<#y<<":
"<y<<" | "<<#z<<": "<<z<<endl;
#define trace4(a, b, c, d)   cout <<#a<<": "<<a<<" | "<<#b<<":
"<b<<" | "<<#c<<": "<<c<<" | "<<#d<<": "<<d<<endl;
#define trace5(a, b, c, d, e) cout <<#a<<": "<<a<<" | "<<#b<<":
"<b<<" | "<<#c<<": "<<c<<" | "<<#d<<": "<<d<<" | "<<#e<<":
"<e<<endl;
#define _F(i,a,b) for(ll i = a ; i<b ; ++i)
#define __F(i,a,b) for(ll i = a ; i>=b ; --i)
#define f1(a) for(ll i = 0 ; i<a ; ++i)
#define f2(a) for(ll j = 0 ; j<a ; ++j)
#define f3(a) for(ll k = 0 ; k<a ; ++k)
#define mp make_pair
#define pb push_back
#define vl vector<ll>
#define vd vector<double>
#define pll pair<ll,ll>
#define vll vector<pair<ll,ll> >
#define vls vector<pair<ll,string> >
#define mll map<ll,ll>
#define msl map<string,ll>
#define F first
#define S second
#define _fast_ ios_base::sync_with_stdio(false),cin.tie(NULL);
#define inf 1e18
#define MOD (ll)(1e9 + 7)
//auto cmp=[](pll const & a, pll const & b){return a.F != b.F ?
a.F<b.F : a.S < b.S;};

void solve(){
    ll m,n;
    cin>>m>>n;
    ld temp = 0;
    ld sum = 0 ;
    ld curr ;
    f1(m){
        curr = (ld)(i+1)/(ld)m;

```

```
        curr = (ld)pow(curr,n);
        sum += ((i+1)*(curr-temp));
        temp = curr;
    }
    cout<<fixed<<sum<<endl;
}

int main(){
    _fast_
    ll t =1;
    //cin>>t ;
    while(t--){
        solve() ;
    }
}
```

