CODE FORCES
Sponsored by Telegram

HOME   TOP   CONTESTS   GYM   PROBLEMSET   GROUPS   RATING   API   CALENDAR   HELP   **10 YEARS!** 🎁

RED_AWHILE   BLOG   TEAMS   SUBMISSIONS   GROUPS   CONTESTS   PROBLEMSETTING
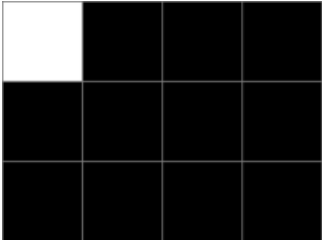
ReD_AwHiLe's blog

Codeforces Round #632 (Div. 2) editorial

By **ReD_AwHiLe**, 31 hour(s) ago, 🇬🇧, 🖉

I hope you liked problems!

Sorry for incorrect placement of problems. I had to do swap(E, F).

1333A - Маленький Артем

In this problem it is enough to simply paint the upper left corner white and all the others black for any size of the board Like this:

And there are $W = 1$ (cell with coordinates $\{1, 1\}$) and $B = 2$ (cells with coordinates $\{1, 2\}$ and $\{2, 1\}$).

In the first version, the task restrictions were $1 \le n, m$, but we thought it would be too difficult for div2A.

Solution C++

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

using namespace std;


void solve() {
    int n, m; cin >> n >> m;
    string black_row(m, 'B');
    vector<string> result(n, black_row);
    result[0][0] = 'W';
    for (int i = 0; i < n; ++i) {
        cout << result[i] << '\n';
    }
}


int main() {
    int t; cin >> t;
    while(t--) solve();
}
```

1333B - Kind Anton

First of all, note that we can add an element with index i to an element with index j iff $i < j$. This means that the element a_n cannot be added to any other element because there is no index $j > n$ in the array. This is why we can first equalize the elements a_n and b_n. If $a_n = b_n$, they are already equal. If $a_n < b_n$, then we need to have element equal to

1 along the elements a with indexes $\{1, \ldots, n-1\}$. For $a_n > b_n$, we need to have -1 along these elements. After the elements with index n become equal, we can go to the element with index $n-1$ and do the same. Then indexes $n-2, n-3, ..., 1$. You can implement this idea yourself!

Final time complexity: $O(n)$

Solution C++

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

using namespace std;

void solve() {
    int n; cin >> n;
    vector<int> a(n), b(n);
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
    }
    for (int i = 0; i < n; ++i) {
        cin >> b[i];
    }
    vector<int> good(2, 0);
    for (int i = 0; i < n; ++i) {
        if (a[i] > b[i] && !good[0]) {
            cout << "NO\n";
            return;
        } else if (a[i] < b[i] && !good[1]) {
            cout << "NO\n";
            return;
        }
        if (a[i] == -1) good[0] = 1;
        if (a[i] == 1) good[1] = 1;
    }
    cout << "YES\n";
}

int main() {
    int t; cin >> t;
    while(t--) {
        solve();
    }
}
```

1333C - Eugene and an array

Let's solve this problem in $O(n^2 \times log(n))$ for now.

Note that if the subarray $[a_i, \ldots, a_j]$ is good, then the subarray $[a_i, \ldots, a_{j-1}]$ is also good, and if the subset $[a_i, \ldots, a_j]$ is not good, then the subarray $[a_i, \ldots, a_{j+1}]$ is not good either. Then for each left border a_i we want to find the rightmost border a_j such that $[a_i, \ldots, a_j]$ is good and add to the answer $j - i + 1$ (subarrays $[a_i, \ldots, a_j], [a_i, \ldots, a_{j-1}], \ldots, [a_i])$ [1]. Let's denote the rightmost border j for border i as $R(i)$.

Let's calculate the prefix-sum of the array P.

$P_0 = 0, P_i = a_1 + \ldots + a_i, 1 \le i \le n.$

Note that a subset of $[a_i, \ldots, a_j]$ has a zero sum iff $P_{i-1} = P_j$. Then the subset $[a_i, \ldots, a_j]$ is a good iff sum of prefixes $[P_{i-1}, \ldots, P_j]$ has no duplicates [2].

Using [1] and [2], we can simply iterate over i from 0 to n and over j from i to n and count the set of prefix sums $[P_i, \ldots, P_j]$. The first moment j_0 when this set contains duplicates gives us the rightmost border $j_0 - 1$, and we add $(j_0 - 1) - i$ (no $+1$, because it is an array of prefix sums) to answer.

To improve this solution to $O(n \times log(n))$, we need to note that $R(i)$ is monotonous over i. Now we can iterate over i from 0 to n and over j from $R(i-1)$ to n uses a set of prefix sums from the previous iteration. Thus we have a solution $O(n \times log(n))$, because j points to each element of the array exactly once.

If you code in C++, it is important not to use std:: unordered_set in this task, but use std::set. One of the participants hacked the solution using std:: unordered_set, using collisions in this structure. I highly recommend you to read this blog for more info https://codeforces.com/blog/entry/62393.

Final time complexity: $O(n \times log(n))$

Solution C++

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

using namespace std;


int main() {
    int n; cin >> n;
    vector<long long> prefix(n + 1, 0);
    for (int i = 0; i < n; ++i) {
        int x; cin >> x;
        prefix[i + 1] = prefix[i] + x;
    }
    int begin = 0, end = 0;
    long long ans = 0;
    set<long long> s = {0};
    while(begin < n) {
        while(end < n && !s.count(prefix[end + 1])) {
            ++end;
            s.insert(prefix[end]);
        }
        ans += end - begin;
        s.erase(prefix[begin]);
        ++begin;
    }
    cout << ans << endl;
}
```

1333D - Досуг в школе №41

If solution exist let's count the minimum and maximum bounds for k for initial arrangement of children. A minimum k achieved all possible pairs of children turn theirs heads at every step. The maximum k reached if only one of possible pairs of children turn theirs heads at every step. This values is easy to count, I'll leave it to you!

If k from the statement not fit within our bounds, then we need to print -1. Otherwise solution exist and we need to construct them. For each next move we can use all pairs of children to turn theirs heads, decrease k by 1 and recalculate maximum bound (lets call it U) on k (just decrease them on the number of pairs used). If after moving new value of k fits in the bound ($k \leq U$), then we proceed to the next iteration. Otherwise, we roll back to the previous iteration and use $U - k + 1$ pairs in this move. Number of remaining moves will be $k - 1$ and upper bound will be $U - (U - k + 1) = k - 1$. And from that moment, just use only one pair in one move to the end of the process (to find one of the pair quickly we need to store them in the queue).

Final time complexity: $O(n^2)$

Solution C++

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

using namespace std;

int n, k;
```

```cpp
vector<int> find_steps(const vector<int>& a) {
    vector<int> steps;
    for (int i = 0; i < n - 1; ++i) {
        if (a[i] == 1 && a[i + 1] == 0) steps.push_back(i);
    }
    return steps;
}


int main() {
    cin >> n >> k;
    string s; cin >> s;
    vector<int> a(n);
    for (int i = 0; i < n; ++i) a[i] = (s[i] == 'L') ? 0 : 1;
    int maxi = 0, mini = 0;
    int cnt = 0;
    int last = -1;
    for (int i = n - 1; i >= 0; --i) {
        if (a[i] == 0) {
            ++cnt;
        } else {
            if (cnt == 0) continue;
            maxi += cnt;
            mini = max(cnt, last + 1);
            last = mini;
        }
    }
    if (k < mini || k > maxi) {
        cout << -1;
        return 0;
    }
    bool is_min = false;
    vector<int> have_step;
    for (int i = 0; i < k; ++i) {
        if (!is_min) {
            auto steps = find_steps(a);
            cout << min(int(steps.size()), maxi - k + i + 1) << ' ';
            int cur = 0;
            while (k - i - 1 < maxi && cur < steps.size()) {
                cout << steps[cur] + 1 << ' ';
                a[steps[cur]] = 0;
                a[steps[cur] + 1] = 1;
                ++cur;
                --maxi;
            }
            if (maxi == k - i - 1) {
                is_min = true;
                have_step = find_steps(a);
            }
        } else {
            int v = have_step.back();
            have_step.pop_back();
            cout << "1 " << v + 1;
            a[v] = 0;
            a[v + 1] = 1;
            if (v > 0 && a[v - 1] == 1) {
                have_step.push_back(v - 1);
            }
            if (v + 2 < n && a[v + 2] == 0) {
                have_step.push_back(v + 1);
            }
        }
        cout << '\n';
```

```
        }
}
```

1333E - Road to 1600

First of all notice that there are no such boards for $N = 1, 2$. Then you can find an example for $N = 3$ by yourself or with counting all cases with program. One of possible examples (I find it using paper, pencil and my hands):

$N = 3$:

9	5	6
7	2	8
1	3	4

For large N we can walk by spiral (like snake) to the case $N = 3$.

$N = 4$:

16	12	13	1
14	9	15	2
8	10	11	3
7	6	5	4

$N = 5$:

25	21	22	10	9
23	18	24	11	8
17	19	20	12	7
16	15	14	13	6
1	2	3	4	5

Rook and Queen first going in a spiral and arrive to $N = 3$ case. It can be used any of such spiral, not just this one.

Final time complexity: $O(N^2)$

Solution C++

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

using namespace std;

int main() {
    int n; cin >> n;
    if (n < 3) {
        cout << -1;
        return 0;
    }
    vector<int> solution = {
        1, 3, 4, 8, 2, 7, 9, 5, 6
    };
```

```
        vector<vector<int>> table(n, vector<int>(n, 0));
    int cur = 1;
    for (int i = 0; i < n - 3; ++i) {
        if (i & 1) {
            for (int j = n - 1; j >= 0; --j) {
                table[i][j] = cur;
                ++cur;
            }
        } else {
            for (int j = 0; j < n; ++j) {
                table[i][j] = cur;
                ++cur;
            }
        }
    }
    if ((n - 3) & 1) {
        for (int j = n - 1; j >= 0; --j) {
            if (j & 1) {
                for (int i = n - 3; i < n; ++i) {
                    if (j > 2) {
                        table[i][j] = cur;
                        ++cur;
                    } else {
                        table[i][j] = solution[(2 - j) * 3 + i - n + 3]
+ n * n - 9;
                    }
                }
            } else {
                for (int i = n - 1; i >= n - 3; --i) {
                    if (j > 2) {
                        table[i][j] = cur;
                        ++cur;
                    } else {
                        table[i][j] = solution[(2 - j) * 3 + n - 1 - i]
+ n * n - 9;
                    }
                }
            }
        }
    } else {
        for (int j = 0; j < n; ++j) {
            if (j & 1) {
                for (int i = n - 1; i >= n - 3; --i) {
                    if (j < n - 3) {
                        table[i][j] = cur;
                        ++cur;
                    } else {
                        table[i][j] = solution[(j - n + 3) * 3 + n - 1
- i] + n * n - 9;
                    }
                }
            } else {
                for (int i = n - 3; i < n; ++i) {
                    if (j < n - 3) {
                        table[i][j] = cur;
                        ++cur;
                    } else {
                        table[i][j] = solution[(j - n + 3) * 3 + i - n
+ 3] + n * n - 9;
                    }
                }
            }
        }
    }
    for (int i = 0; i < n; ++i) {
```

```
        for (int j = 0; j < n; ++j) {
            cout << table[i][j] << ' ';
        }
        cout << '\n';
    }
}
```

1333F - Kate and imperfection

Let $A = \{a_1, a_2, \ldots, a_k\}$ be one of the possible subsets with smallest imperfection. If for any number a_i in A not all of its divisors contained in A then we can replace a_i with one of it divisor. The size os the subset does not change and imperfection may only decrease. Then we can assume that for any a_i all of it divisors contained in A. Let $d(n)$ be the greatest divisor of n exclude n ($d(1) = 1$). Since A contains element with its divisors then smallest gcd of pair of an elements not less than maximum of $d(a_i)$ over elements of A (because A contains a_i with $d(a_i)$). And for any element a_i there is no element $a_j < a_i$ in A with $gcd(a_i, a_j) > d(a_i)$ (because $d(a_i)$ is the greatest divisor). Then imperfection of A is equal to greatest $d(a_i)$ over elements of A. After this observation we can just sort elements $\{1, \ldots, n\}$ by theirs $d(*)$ and take smallest k for every $2 \le k \le n$. You can calculate $d(*)$ using the sieve of Eratosthenes.

Final time complexity: $O(n \times log(n))$

Solution C++

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

using namespace std;


vector<int> max_div;

void eratosthenes(int limit) {
    max_div.assign(limit + 1, 0);
    max_div[0] = limit + 10;
    max_div[1] = 1;
    for (int i = 2; i <= limit; ++i) {
        if (max_div[i]) continue;
        for (int j = i; j <= limit; j += i) {
            if (max_div[j]) continue;
            max_div[j] = j / i;
        }
    }
}


int main() {
    int n; cin >> n;
    eratosthenes(n);
    sort(max_div.begin(), max_div.end());
    for (int i = 1; i < n; ++i) {
        cout << max_div[i] << ' ';
    }
}
```

Tutorial of Codeforces Round #632 (Div. 2)

632,   #editorial

▲ **+111** ▽   ☆   👤 ReD_AwHiLe   🕐 31 hour(s) ago   💬 143

💬 Comments (143)   Write comment?

30 hours ago # topic has been updated by ReD_AwHiLe (previous revision, new +10

ReD_AwHiLe

Auto comment: topic has been updated by ReD_AwHiLe (previous revision, new revision, compare).

→ Reply

---

30 hours ago, # | ☆   ▲ +11 ▼

Finally !! thanks for the editorial.

→ Reply

Tman

---

30 hours ago, # | ☆   ▲ 0 ▼

Can anyone help why i am getting wrong answer on testcase 7 -76004123

→ Reply

coolshubham

---

30 hours ago, # ^ | ☆   ← Rev. 4   ▲ +4 ▼

Read the checker's comment. It is said in the question to print those indices where heads will turn from R to L.

Check this Testcase:
6 6
RLRLRL

→ Reply

shubhammitt

---

29 hours ago, # ^ | ☆   ← Rev. 2   ▲ 0 ▼

I did few changes(but dont know whats the difference between this and previous code) but still not able to pass testcase 7-76027745

→ Reply

coolshubham

---

29 hours ago, # ^ | ☆   ▲ 0 ▼

I got my error. Thanks a lot !!

→ Reply

coolshubham

---

17 hours ago, # ^ | ☆   ▲ 0 ▼

Hi, I've the same problem in my code as well, after taking few hints from this discussion, I made some changes, only to get a different error. Could you help me with this?

→ Reply

I_Love_Totoro

---

30 hours ago, # | ☆   ← Rev. 2   ▲ +63 ▼

I guess many people print something like chess-board on the problem A, so do I lol

Anyway, the editorial of the problem A is really amazing!

→ Reply

fried-chicken

---

29 hours ago, # ^ | ☆   ▲ +15 ▼

i did the same like chess...,but figured out the correct answer after the contest(same as editorial)..baaah!!.

→ Reply

jitto98

---

29 hours ago, # ^ | ☆   ▲ +20 ▼

Your answer is correct too :)

→ Reply

ReD_AwHiLe

---

21 hour(s) ago, # ^ | ☆   ▲ 0 ▼

may be the case B=W+1,made us think like that,rather than the simple way,where we can simply cancel out.

→ Reply

jitto98

---

20 hours ago, # ^ | ☆   ▲ 0 ▼

I am still not getting about B=W+1. What

I am still not getting about B=W+1. What is mean by B=W+1 in problem div2 A. It made me totally confusing!!

dinesh9636286205   → Reply

18 hours ago,   # ^ | ☆   ▲ 0 ▼

the difference between no of blacks with atleast 1 adjacent white and no of white with at least 1 black as adjacrent is 1.

jitto98   → Reply

new, 6 hours ago,   # ^ | ☆   ▲ 0 ▼

if the chess board is correct, then why it was showing wrong answer.

→ Reply

iam_anurag_9

new, 5 hours ago,   # ^ | ☆   ▲ 0 ▼

like chess board, but a little difference.

→ Reply

ReD_AwHiLe

new, 4 hours ago,   # ^ | ☆ ▲ 0 ▼

What I did is if n*m is even then make a chess board wont statisfy b=w+1 so I put B in last in place of white to satisfy the condition. if odd, then just put BWBW..... till last. I wonder where I am wrong.

→ Reply

iam_anurag_9

16 hours ago,   # ^ | ☆   ▲ -7 ▼

I think the question should be like B>=W+1 not B=W+1.

→ Reply

jakaria900

16 hours ago,   # ^ | ☆   ▲ 0 ▼

This looks like actual question

→ Reply

dinesh9636286205

9 hours ago,   # ^ | ☆   ▲ 0 ▼

no coz B means number of black boxes who have white neighbors not all the blacks

→ Reply

codegirl22

30 hours ago,   # | ☆   ← Rev. 2   -31 ▼
→ Reply

The comment is hidden because of too negative feedback, click here to view it

ma_da_fa_ka

29 hours ago,   # ^ | ☆   ▲ +36 ▼

Consider n = 9

mightymercado

Of course, we first take all the prime number from 2 to 9 and 1 as well.

| 1 | 2 | 3 | 5 | 7 |

Hence, for k = 2 to 5, the minimal imperfection is 1 because the gcd is always 1.

For k = 6, we pick 4 so minimal imperfection is now 2.

For k = 7, we pick 6 so minimal imperfection is now 3.

For k = 8, we pick 9 so minimum imperfection is still 3.

For k = 9, we pick 8 so minimum imperfection is now 4

For k = 5, we pick 8 so minimum imperfection is now 4.

In this example, it doesn't hold that we take the smallest composite number every time. Correct me if im wrong.

→ Reply

---

30 hours ago, # | ☆   +18

The problems were very good but the tutorial is too damn late bro still you really did a good job thanks!

→ Reply

Abhayp001

---

30 hours ago, # | ☆   0

Can someone please explain why my solution for problem A is failing. 76024216 Maybe, there's an issue with my declaration of the 2d array, as the output to some of the subcases has 'W' in the top left as well as right corner which is not what I intend to do.

→ Reply

falconlover

---

29 hours ago, # ^ | ☆   ← Rev. 2   0

Your array is initially filled with junk values (values that are uninitialised i.e. initialised to whatever junk was present in that memory location previously). So, any particular value may or may not be equal to 1 (or any number as a matter of fact). Make your array declaration global of sufficiently large size (or use a vector and resize it for different input) (or manually assign all values in C-style array to 0 for every input N,M)

→ Reply

Zeus_NoT_Zues

---

29 hours ago, # ^ | ☆   ← Rev. 2   0

just declarate array before main and u will get AC.and if u will declarate it before main u not need to initialize array by zero

→ Reply

Ebiarat

---

29 hours ago, # ^ | ☆   0

And how say Yuki726 "You are using i for iterating tests and rows both."

→ Reply

Ebiarat

---

29 hours ago, # ^ | ☆   0

That would solve it. But my question is why is my code failing? Isn't declaring the array as int arr[n][m]={0} not the correct way of declaration?

→ Reply

falconlover

---

29 hours ago, # ^ | ☆   0

That will only make arr[0][0] initialise to 0.... Not all of arr[0...N — 1][0...M — 1] like you expect it to.

→ Reply

Zeus_NoT_Zues

---

29 hours ago, # ^ | ☆   ← Rev. 2   0

You are using i for iterating tests and rows both.

→ Reply

Yuki726

---

29 hours ago, # ^ | ☆   0

Changing variable names didn't help 76026831.

→ Reply

falconlover

---

29 hours ago, # ^ | ☆   ← Rev. 2   0

Here, an accepted version of your code...

Edit: Like I mention in the comment below (reply to Yuki726), your variable declarations don't cause an

Zeus_NoT_Zues

issue. Even if they were to, it'd be a compile time

---

issue. Even if they were to, it'd be a compile time error.
→ Reply

29 hours ago, # ^ | ☆   ▲ +6 ▼

That wouldn't cause the issue. Even if it does, it should be a compile error. The access to arr[i][j] will use the closest declared 'i' in the scope (similar to how you can shadow a global variable of the same name inside a function).

Zeus_NoT_Zues → Reply

29 hours ago, # ^ | ☆   ▲ +6 ▼

ok, my above answer is not correct. initializing array with ={0} is not correct, try initializing with ={} and it passes, 76027021
→ Reply

Yuki726

28 hours ago, # ^ | ☆   ← Rev. 2   ▲ +1 ▼

I believe `= {0}` syntax is only valid for statically allocated arrays. Though I cannot find a reference for this. It doesn't even compile on my GCC.

mightymercado → Reply

29 hours ago, # | ☆   ▲ +11 ▼

Do anyone know how to prove minimum bound for k in problem.D?
→ Reply

Yuki726

29 hours ago, # ^ | ☆   ← Rev. 3   ▲ +11 ▼

Using dynamic programming, you can calculate for each 'R' what the maximum number of moves it will take to get its position in the final array. Formula:

dp[cur 'R'] = max(dp[next 'R'] + 1,'L' cnt on [i + 1, ..., n]).

It only remains to note that for the strategy in the solution, this estimate is achieved.

ReD_AwHiLe → Reply

29 hours ago, # ^ | ☆   ▲ 0 ▼

I just spoilered how to calculate this value :(
→ Reply

ReD_AwHiLe

29 hours ago, # ^ | ☆   ▲ +16 ▼

Thanks for the reply. Great contest btw.
→ Reply

Yuki726

29 hours ago, # | ☆   ▲ 0 ▼

can anyone help me? It says i failed on test2 token 175: 76028180
→ Reply

abhinav12

29 hours ago, # ^ | ☆   ▲ 0 ▼

probem 2*
→ Reply

abhinav12

29 hours ago, # ^ | ☆   ▲ 0 ▼

Your logic is wrong. Sub-testcase-175 of TestCase-2, your code prints "YES" whilst the answer is "NO"
→ Reply

Zeus_NoT_Zues

28 hours ago, # ^ | ☆   ▲ 0 ▼

1

Yuki726

```
1
2
-1 -1
-1 1
```

try this case.

→ Reply

---

15 hours ago, # ^ | ☆   ▲ 0 ▼

Solved Thanks!

→ Reply

abhinav12

---

28 hours ago, # | ☆   ▲ 0 ▼

Is it possile to solve C using segment tree.If yes then please give your code or explain it.

→ Reply

Hamim99

---

28 hours ago, # | ☆   ▲ +4 ▼

Somebody want to explain F? I do not get it at all. What is the question, what is the idea to solve it?

There is obviously something with divisors because the imperfection is defined using gcd. And then?

→ Reply

spookywooky

---

13 hours ago, # ^ | ☆   ▲ +4 ▼

The problem: Given the set of integers from 1 to n, $\forall\, i \,\epsilon\, [2..n]$, find a set, M of size i such that max value of $\gcd(p, q) \,\forall\, p, q \,\epsilon\, M$ is minimized.

The solution: Firstly, note that 1 must always be included in the optimal set. Next, note that if there are P primes from $[2..n]$, then max value of $\gcd(p, q) \,\forall\, p, q \,\epsilon\, M$ is always 1. Hence, for the first P numbers, answer will always be 1. Now from the $p + 1$ th number onwards, if we include an integer x, the optimal set always contains at least 1 divisor d $(1 \leq d < x)$ of x. Why? Since we have already included all the primes from $[2..n]$, all the numbers we are left with are composite. Now if we add x to the set M, max value of $\gcd(p, q) \,\forall\, p, q \,\epsilon\, M$ is the largest divisor of x, since $\gcd(\text{largest divisor of } x, x) = \text{largest divisor of } x$. Hence, we have to add elements to set greedily such that the largest divisor of $x \,\forall\, x \,\epsilon\, M$ is minimized. For this, we can find out the largest divisor, $d \,\forall\, i \,\epsilon\, [2..n]$, and store it in a list v. The answer is the sorted list v.

Code

→ Reply

D1703

---

new, 7 hours ago, # ^ | ☆   ▲ 0 ▼

why we are adding the greatest divisor in set instead we should add the smallest divisor to minimize perfection?

→ Reply

harshitsinghal.be18cse

---

28 hours ago, # | ☆   ← Rev. 2   ▲ 0 ▼

In Second problem In this test case may be editorial's answer is wrong in

1

4

0 2 4 6

0 2 8 10

I have mad b from a. Like, 3rd element of b is made from 2 time add 2nd element of a, etc. I have also checked others code some of them gave answer as Yes, Other gave No .

If I am wrong, please reply me

Neel_Patel

If I am wrong, please reply me.
→ Reply

28 hours ago, # ^ | ☆ 0

ReD_AwHiLe

Array a consists only of {-1, 0, 1}
→ Reply

28 hours ago, # ^ | ☆ 0

Thanks
→ Reply

Neel_Patel

28 hours ago, # ^ | ☆ 0

AadityA_Singh

array a can contain elements -1,0,1 only so your test case is incorrect
→ Reply

28 hours ago, # ^ | ☆ 0

Thanks
→ Reply

Neel_Patel

28 hours ago, # | ☆ ← Rev. 2 0

Can someone explain why in problem A, for testcase 2 BWB WBW BWB is not an accepted solution

Gilfoyl

https://codeforces.com/contest/1333/submission/75973141
→ Reply

28 hours ago, # ^ | ☆ +11

because in your output: BWB BBB there three B where a W is of them(2 in first row,1 in middle of second row).So B=3 here.Again a W has a adjacent B.So w=1.It doesn't maintain B=W+1.

Hamim99

→ Reply

28 hours ago, # ^ | ☆ 0

thanks for pointing out my error
→ Reply

Gilfoyl

28 hours ago, # ^ | ☆ 0

Welcome.
→ Reply

Hamim99

28 hours ago, # | ☆ ← Rev. 6 +16

Here is how I up-solved D. Spoiler, it is long to be newbie friendly.

Note: if you are getting TLE in this problem. Use '\n' instead of `endl` to enabled output buffering because `endl` will flush the output. You don't want to flush the output everytime you need to output a newline because that means doing IO. The TL is very tight.

mightymercado

Spoiler
→ Reply

28 hours ago, # ^ | ☆ 0

Excellent explanation!
→ Reply

sh_maestro

17 hours ago, # ^ | ☆ 0

Dude... This is giving TLE..
→ Reply

risk_17

15 hours ago, # ^ | ☆ 0

See my submission. I didnt get TLE. And I checked your

See my submission. I didn't get TLE. And I checked your submission, it has many differences such that you are performing the swaps right away

→ Reply

mightymercado

13 hours ago,   # ^ | ☆   ▲ 0 ▼

does it matter if we perform the swaps right away. even i am getting TLE

→ Reply

teamcaproxx

12 hours ago,   # ^ | ☆ ← Rev. 3   ▲ 0 ▼

I'm not saying performing swaps right away will (necessarily) cause TLE. I'm just saying there's a lot of differences in my solution to **risk_17**'s. And YES, it would matter.

Consider `RRLL`

How many swaps can you simultaneously in the first step?

mightymercado

Answer: 1

But if you perform swaps right away:

`RRLL` becomes `RLRL` then becomes `RLLR` in one loop. So, first, find all `L`'s preceded by an R. Then only AFTER, swap them.

→ Reply

teamcaproxx

12 hours ago,   # ^ | ☆   ▲ 0 ▼

yes i have taken care of what you are saying but still i get tle

→ Reply

12 hours ago,   # ^ | ☆   ▲ 0 ▼

You are right, it seems weird that you are getting TLE. After checking, using `\n` instead of `endl` will get you AC. I think the TL is very tight and the output is huge, so input buffering must be enabled.

mightymercado

Here is your solution modified to get AC: https://codeforces.com/contest/1333/submission/76085125

→ Reply

11 hours ago, 0,
^ | ☆

thanks

→ Reply

teamcaproxx

10 hours ago,   # ^ | ☆   ▲ 0 ▼

Please check it.. I had incremented i value if any swaps is there..

→ Reply

risk_17

10 hours ago,   # ^ | 0 ☆

You are right. You can perform the swaps right away if you increment twice during any given swap :)

mightymercado

→ Reply

new, 9 hours ago, # ^ |

risk_17

OK...After changing endl to \n its not giving TLE.. maybe because of the large amount of output

→ Reply

28 hours ago, # | ▲ 0 ▼

The problem A says that B should be equal to W+1 but in the tutorial number of W is always 1 then how it is a good coloring what am i missing??Please help..

AadityA_Singh   → Reply

28 hours ago, # ^ | ▲ +1 ▼

There are 2 black cells with a border to the white one, and the one white one.

So B=2, W=1

spookywooky   → Reply

28 hours ago, # ^ | ▲ 0 ▼

oh thanks,got it

→ Reply

AadityA_Singh

28 hours ago, # ^ | ▲ 0 ▼

B is not equal to total number of B. B is equal to total number off such B whice has a adjacent W. I hope you understand.

→ Reply

Hamim99

28 hours ago, # | ▲ +3 ▼

What is the time complexity of this solution for problem C, by **tmwilliamlin168**?

→ Reply
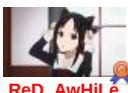
kingmanas

28 hours ago, # ^ | ▲ +6 ▼

This solution is similar to the solution at editorial. So nlogn

→ Reply

ReD_AwHiLe

28 hours ago, # ^ | ▲ 0 ▼

Thank you for replying so quickly

→ Reply

kingmanas

28 hours ago, # | ▲ 0 ▼

Can anyone help me to understand error in my code for (133C) my code

→ Reply

pk_2809

21 hour(s) ago, # ^ | ▲ 0 ▼

Your code give 3 for this case, but it should give 2.

```
3
1 0 -1
```

→ Reply

sare

18 hours ago, # ^ | ▲ 0 ▼

Thanks, I forgot to keep the 'st' pointer to rigtmost position

→ Reply

pk_2809

28 hours ago, # | ☆   ▲ **0** ▼

Explain C in simple language and example.
→ Reply

time_14_07

17 hours ago, # ^ | ☆   ← Rev. 2   ▲ **+3** ▼

In C we were asked to count number of subarrays that were good (i.e
their subarrays must not have zero sum)

Now few observations before starting : 1. If a subarry is good that
means all of 'its' subarrays are also good. 2. If a subarray is not good
then all subarrays which will contain this subarray is also not good.
example: 1 -1 3 4 now sum(1 -1) = 0 so it is a bad subarray so if you
include (1 -1 3) this is also bad for same reason and so is (1 -1 3 4)

now main problem is to find subarrays with 0 sum in efficient manner for
which we use idea of prefix sum. what we try to do is find sum of all
elements coming before it. for example 1 2 3 4 is the array then it's
prefix sum is 1 3 6 10. now we can say that there exist a subarray with
0 sum if : 1. prefix itself is 0. 2. prefix in some index is seen before.

RajatBansal16

let's say 1 2 -3 4 -1 it's prefix sum is 1 3 0 4 3 so there are two cases
where subarray is 0 one is (1 2 -3) as prefix index was 0 here and
second is (-3 4 -1) as prefix 3 in last index was seen before in 3rd
index. we can use maps to find subarray with 0 sum to efficiently. Hope
that helps.
→ Reply

28 hours ago, # | ☆   ▲ **0** ▼

I'm a bit confused right now. So you advise us to use set/map instead of
unordered_map/unordered_set so we don't get TLE, because some "adorable
community colleagues" added test cases to make it non-viable.

Does this recommendation stand only for this particular problem or we will have
to use it from now on? Won't we get TLE on other problems for using an ordered
data structure when it isn't generally needed?
→ Reply

Tirutu

27 hours ago, # ^ | ☆   ← Rev. 2   ▲ **+5** ▼

Unordered set and unordered map use hashing, so if someone creates
anti-hashing testcases ordered data structures are faster. In the worst
case data structures that use hashing take O(N) complexity to access a
element where N is the number of element in the data structure.
Ordered data structures like set and map always have a O(logN)
complexity. Use ordered data structures to decrease the risk of getting a
TLE.

filippodigra

Sorry for my bad english.
→ Reply

26 hours ago, # ^ | ☆   ▲ **0** ▼

What actually concerns me is: could it happen to get TLE
using ordered map/set but pass with unordered(in case
someone doesn't make anti-hash test cases)??
→ Reply

Tirutu

25 hours ago, # ^ | ☆   ▲ **0** ▼

No. Just do not use unordered_set/map without
modification of hashing.
→ Reply

spookywooky

28 hours ago, # | ☆   ▲ **+1** ▼

I solved C using recursion. 75900316
→ Reply

yash_daga

27 hours ago, # | ☆   ▲ **0** ▼

The answer to the first problem is like a joke :) Still very clever. This was my first

kilicars

The answer to the first problem is like a joke :) Still very clever. This was my first contest in Codeforces and I could only submit this problem and got wrong answer and spent most of the time searching for a test case breaking my code but I couldn't and gave up. And now I noticed that I put a space between Bs and Ws. I also noticed that I can see why the test was not accepted. My question is, could I see that during the contest as well? Thanks.

→ Reply

27 hours ago, # | ☆   ▲ 0 ▼

Hamim99

In this tutorial,for problem C,Isn't the loop is amortize?Why not complexity is O(n+n)? Please anyone explain.

→ Reply

27 hours ago, # | ☆   ▲ 0 ▼

include <bits/stdc++.h>

using namespace std;

long long n; map<long long,long long> ls;

int main() { long long i,k,sm=0,mx=0,z=0;

```
scanf("%lld",&n);
ls[0]=1;
for(i=1;i<=n;i++)
{
    scanf("%lld",&k);
    sm+=k;
    mx=max(mx,ls[sm]);
    z+=i-mx;
    ls[sm]=i+1;
}
printf("%lld\n",z);

}
```

lazy_learner

This works for problem C but what I don't understand is why do we have

mx=max(mx,ls[sm]); Is this not equivalent to **if(ls.find(sm)!=ls.end())mx = ls[sm];**

→ Reply

27 hours ago, # ^ | ☆   ▲ 0 ▼

Got it.

→ Reply

lazy_learner

26 hours ago, # ^ | ☆   ▲ +1 ▼

Code in `Spoiler` , please?

→ Reply

Zeus_NoT_Zues

26 hours ago, # | ☆   ← Rev. 2   ▲ 0 ▼

In problem C what should be the output of 1 2 0? for me it should be output-> 5 (1),(2),(1,2)(1,2,0),(2,0) but editorial solution gives 3

→ Reply

Cortex_guest

26 hours ago, # ^ | ☆   ▲ 0 ▼

0's should be excluded, so answer is 3

→ Reply

sh_maestro

25 hours ago, # ^ | ☆   ▲ 0 ▼

Ok got it thanks for reply

→ Reply

Cortex_guest

26 hours ago, # ^ | ☆   ← Rev. 2   ▲ 0 ▼

Firstly, how is {1,0} a sub-array of {1, 2, 0}?

Zeus_NoT_Zues

Also, {2,0} is incorrect because its sub-array {0} is not valid. So, answer is 3.

EDIT: (following your edit) {1,2,0} is also not valid, as its sub-array {0} is invalid.

→ Reply

25 hours ago, # ^ | ☆   ▲ 0 ▼

Ok got it thanks for reply
→ Reply

Cortex_guest

26 hours ago, # | ☆   -17 ▼

Plz, write editorial in a more descriptive way. Write by assuming we don't know, not just write by assuming you are revising something, although this is for only last question's editorial.

nishant_raja

→ Reply

21 hour(s) ago, # ^ | ☆   ▲ 0 ▼

The editorials are already well descriptive.
You will cope with understanding them gradually.
Keep reading until you understand and seek for help if you need.
And be patient.

MubtasimShahriar

→ Reply

25 hours ago, # | ☆   ▲ 0 ▼

I tried to implement C before check author's code, but my implementation is wrong, and I don't see difference between my and author's code, can u help me find where is my solution going wrong

Code

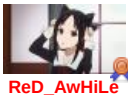**N0N4M3**   → Reply

23 hours ago, # ^ | ☆   ← Rev. 2   ▲ +1 ▼

set dup is empty at the start of every iteration by i. Also you need to use long long instead of int.

ReD_AwHiLe   → Reply

22 hours ago, # ^ | ☆   ▲ +4 ▼

I do some fixes. Check it
https://codeforces.com/contest/1333/submission/76047566

ReD_AwHiLe   → Reply

12 hours ago, # ^ | ☆   ▲ 0 ▼

I got it. Thanks a lot!
→ Reply

N0N4M3

23 hours ago, # | ☆   ← Rev. 2   ▲ 0 ▼

Can somebody please help? I'm getting wrong answer on test #8 for problem D
— 76047171

black_immortal   → Reply

22 hours ago, # | ☆   ▲ 0 ▼

Can anyone explain me the editorial's logic for question 2??
→ Reply

arsalan9911

20 hours ago, # ^ | ☆   ▲ 0 ▼

Important points 1) array a consists of {-1, 0, 1} 2) pair(i,j) while i < j add ai to aj. 3) to make ai become bi, we need value 1 for ai < bi while value

TypicalNub

-1 for ai > bi However based on point 2 i < j mean that we can only

-1 for ai > bi However based on point 2, i < j, mean that we can only consider the value appear before the current index i. For example a = {0, -1, 1} b = {0, 1, -1} at index 2, ai = -1, bi = 1, we can only consider the values of array a that appear before index 2, in this case is 0, hence we cannot make ai become bi as we need value 1 since ai < bi

→ Reply

19 hours ago, # ^ | ☆   ▲ 0 ▼

thnkx bro

→ Reply

arsalan9911

18 hours ago, # ^ | ☆   ▲ 0 ▼

No problem, hope it helps

→ Reply

TypicalNub

21 hour(s) ago, # | ☆   ▲ 0 ▼

Hello, I got WA on pretest 9 for problem Div2C. Can anyone please tell me what is pretest 9? My submission. Any help will be greatly appreciated.

→ Reply

nadeemshaikh

19 hours ago, # | ☆   ▲ 0 ▼

Can anyone please explain the editorial of C, specially the first solution with O(n^2 logn) solution.Here it is said that if a subarray with [ai.....aj] is good then [ai....aj-1] is also good, i'm not clear how it works.if i take an array of {1, 2, -3, 1, 0} then a subarray {1, 2, -3, 1} is good but {1, 2, -3(aj-1)} is not good, or i misunderstood ? Plz help anyone

→ Reply

LegendaryNewbie_

18 hours ago, # ^ | ☆   ▲ 0 ▼

here is an explanation, hope it will help you.

→ Reply

Tman

17 hours ago, # | ☆   ▲ 0 ▼

Problem D is extension of BINARY MOVEMENTS. I will show how. Let us assume person looking to Right as **0** and Left as **1**. We want to swap these values so the they apart to each other, means change all 01 to 10 state.So in the end our array will look like 111...0000. This problem can even be solved in *O(n)* .If i am wrong please point it out as I am a beginner.According to the editorial of the problem. I will copy the editorial here.

Explaination of Problem Setter of the other problem

→ Reply

sumantopal07

17 hours ago, # | ☆   ▲ 0 ▼

can anyone explain problem A?

→ Reply

gyan_p_t

16 hours ago, # ^ | ☆   ← Rev. 2   ▲ **+1** ▼

Take an NxM board, it is allowed to color a cell as black or white. If a black cell has atleast one adjacent white cell, lets increment variable B. Same holds for W.

The task asks to color such that, B = W + 1. Now, the immediate way that strikes is to color like a chess board. This has one corner case to handle: when value of NxM is even, then we have B == W. So, we have to color one more white spot as black, preferably the position: 0,0 [Since I've assumed coloring all odd sum cells as black. My soln. for reference: link

The editorial provides a much simpler implementation, as in, color only the top-left cell as white and the rest as black. Why is this correct? Well, we have W = 1, since there's only one white cell, also, it has 2 neighboring black cells. We also have B = 2 always, this is because only two cells: (0,1) and (1,0) have neighboring white cell (other blacks only have black neighbors). Thus, the constraint: W = B + 1, always holds. I would recommend visualising/drawing out and seeing. Bonus: As the editorial mentions, the problem gets a little tricky if 1 <=

I_Love_Totoro

Bonus: As the editorial mentions, the problem gets a little tricky if 1 <=
N,M! I realised why only after typing out this explanation.
→ Reply

17 hours ago, # | ☆   ▲ 0 ▼

Can someone explain eugene one. I am unable to get it. I was relating this to
count arrays with sum zero and using that approach. and then subtracting
n(n+1)/2-count.

But I am not able to figure out how to include in count those subarrays of the
given array whose subarrays are also not good.
→ Reply

Hritwik_gok

15 hours ago, # ^ | ☆   ▲ 0 ▼

You can check my solution here
→ Reply

mukesh_g7

15 hours ago, # | ☆   ▲ 0 ▼

In Problem C, Understood all before

**"we need to note that R(i) is monotonous over i. Now we can iterate over i
from 0 to n and over j from R(i−1) to n uses a set of prefix sums from the
previous iteration"**

Harish08

Can anyone help me with this part? An example would be really helpful.
→ Reply

11 hours ago, # ^ | ☆   ▲ 0 ▼

Let's say for i, j is the rightmost position for making a good subarray.
Then when consider i + 1, (i+1,...j) is also good. Hence we can check j
from j + 1
→ Reply

Haochao

15 hours ago, # | ☆   ▲ +9 ▼

The spiral path idea in E is so nice!!
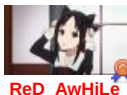→ Reply

aditya_sheth

15 hours ago, # | ☆   ← Rev. 3   ▲ +5 ▼

For given constraints we do not need a sieve in problem F. We can find smallest
prime divisor with naive brute force and get AC, 187 ms
→ Reply

dmkozyrev

11 hours ago, # ^ | ☆   ▲ 0 ▼

Do you know the asymptotics of your solution (without sorting)? Its look
like n*e operation so O(n), but I'm not sure.
→ Reply

ReD_AwHiLe

10 hours ago, # ^ | ☆   ← Rev. 2   ▲ 0 ▼

It is $O(n\sqrt{n})$

UPD. Looks like time complexity is $O\left(\dfrac{n\sqrt{n}}{\log n}\right)$
→ Reply

dmkozyrev

10 hours ago, # ^ | ☆   ▲ 0 ▼

Why?
→ Reply

ReD_AwHiLe

10 hours ago, # ^ | ☆   ▲ 0 ▼

We can run this function for some n and

dmkozyrev

We can run this function for some n and calculate $\dfrac{n\sqrt{n}}{\#\text{operations}}$. This is experiment.

Looks like time complexity is $O\left(\dfrac{n\sqrt{n}}{\log n}\right)$

→ Reply

---

new, 5 hours ago, # ^ | ☆ ▲ **0** ▼

ReD_AwHiLe

Yes, it's looks like O(n sqrt(n) / log(n)), but it's interesting to find accurate complexity. New challenge!

→ Reply

---

13 hours ago, # | ☆ ▲ **0** ▼

Rafiqul01

In problem D.

If I print just only one pair in one move to the start of the process to make time equal to k. what will happen then?

→ Reply

---

13 hours ago, # | ☆ ← Rev. 2 ▲ **0** ▼

Auser1

Can anyone help getting WA on test 76
https://codeforces.com/contest/1333/submission/76074501

→ Reply

---

13 hours ago, # | ☆ ← Rev. 4 ▲ **+3** ▼

thongnon1996

Just want to suggest that an O(n) solution of problem C is possible.

Python solution: http://codeforces.com/contest/1333/submission/76077853

→ Reply

---

10 hours ago, # ^ | ☆ ▲ **0** ▼

Orion

Yes, I also solved for O(n). Here is my C++ solution:
https://codeforces.com/contest/1333/submission/75888071

→ Reply

---

12 hours ago, # | ☆ ▲ **0** ▼

Rafiqul01

I cant find the error of this code. 76082322

can anyone please give me any testcase??

problem D.

→ Reply

---

12 hours ago, # | ☆ ▲ **0** ▼

1_23

please explain me the following lines separately, I cant understand

void solve() { int n, m; cin >> n >> m; string black_row(m, 'B'); vector result(n, black_row); result[0][0] = 'W'; for (int i = 0; i < n; ++i) { cout << result[i] << '\n'; } }

→ Reply

---

12 hours ago, # | ☆ ▲ **0** ▼

jerry_30

for the first problem how it satisfy the condition if there is only white cell the left corner , since B=W+1 and for 3,2 then it will be 5=1+1 and also the adjacent side is not of opposite colour

→ Reply

---

12 hours ago, # | ☆ ▲ **0** ▼

dinesh9636286205

include <bits/stdc++.h>

typedef long long ll; using namespace std;

typedef long long ll; using namespace std;

int main() { long long n; cin>>n; long long arr[n]; long long prefix=0; for (long long i=0;i<n;i++){ cin>>arr[i]; } long long ans=0; bool flag=true; for(long long i=0;i<n;i++){ if(arr[i]==0)continue; for(long long j=i;j<n;j++){ prefix+=arr[j]; if(arr[i]==0)break; if(prefix==0){ans+=j-i;flag=false;prefix=0;break;} } if(flag)ans+=n-i; flag=true; } cout<<ans<<endl;

}

→ Reply

---

12 hours ago, # | ☆   ▲ 0 ▼

Why my code doesn't have n2logn time complexity. I stopping the second loop wherever i am finding the sum of subarray is zero. please help me ?

dinesh9636286205   → Reply

---

11 hours ago, # | ☆   ▲ 0 ▼

I have some problems with E, according to the author's solution for N = 3 case. rook goes through cells: 1 -> 3 -> 2 -> 5 -> 6 -> 4 -> 8 -> (1 vun) -> 7 -> 9 pays 1; Queen goes through cells: 1 -> 2 -> 3 -> 4 -> 6 -> 5 -> 7 -> 9 -> (1 vun) -> 8 pays 1; The cost is equal...... Is there something wrong with my path ?

sywVivian   → Reply

---

10 hours ago, # | ☆   ← Rev. 3   ▲ 0 ▼

ReD_AwHiLe In your solutions for 1333D - Challenges in school №41 there is a small mistake in computing mini(minimum possible k).

It's giving wrong answer for the case-

6 2

RLLRRR

ajay.07   → Reply

> **new**, 9 hours ago, # ^ | ☆   ▲ 0 ▼
>
> hey **ajay.07**, may you help me to know how can we determine min and max numbers of operations
>
> **imasnegi**   → Reply

> **new**, 9 hours ago, # ^ | ☆   ▲ 0 ▼
>
> Thank you! Fixed.
>
> **ReD_AwHiLe**   → Reply

---

new, 9 hours ago, # | ☆   ▲ 0 ▼

May anyone explain how did she find min and max value in problem D?

imasnegi   → Reply

---

new, 9 hours ago, # | ☆   ▲ 0 ▼

*Auto comment: topic has been updated by **ReD_AwHiLe** (previous revision, new revision, compare).*

ReD_AwHiLe   → Reply

---

new, 4 hours ago, # | ☆   ▲ 0 ▼

Hello, Can someone please explain C question duplicate prefix sum part of the tutorial? I didn't understand why that should be true?

Thanks in advance!

cs18b012   → Reply

> **new**, 3 hours ago, # ^ | ☆   ▲ 0 ▼
>
> just a simple thing you have to maintain prefix array and then if any value repeats that show that the sum of elements including this indices is zero than you can easily calculate how much subarrays contains this subarrays after counting all subarrays which contain subarray with sum=0 subtract it from total number of subarrays that can be found by
>
> **sjd0**

sum=0 subtract it from total number of subarrays that can be found by n*(n+1)/2

→ Reply

new, 3 hours ago, # ^ | ☆ ▲ 0 ▼

Thanks for the explanation, but can you explain why duplicates in a range (as mentioned in the tutorial) will create problem?

cs18b012

→ Reply

new, 3 hours ago, # | ☆ ← Rev. 4 ▲ 0 ▼

can anyone help me please, actually I think my code doesn't have a mistake but it fails on the test case 3 how can I improve this code
https://codeforces.com/contest/1333/submission/76197328

2powerofn

→ Reply

new, 3 hours ago, # ^ | ☆ ▲ 0 ▼

this code can be done in O(n). you don't need to actually change the value in first array you can just check is it possible to make it equal to value at same index in second array

sjd0

→ Reply

↑
15
↓