

Contest Duration: 2025-05-24(Sat) 08:00 (<http://www.timeanddate.com/worldclock/fixedtime.html?iso=20250524T2100&p1=248>) - 2025-05-24(Sat) 09:40 (<http://www.timeanddate.com/worldclock/fixedtime.html?iso=20250524T2240&p1=248>) (local time) (100 minutes)

[Back to Home \(/home\)](/home)

[🏠 Top \(/contests/abc407\)](/contests/abc407) [☰ Tasks \(/contests/abc407/tasks\)](/contests/abc407/tasks)

[❓ Clarifications \(/contests/abc407/clarifications\)](/contests/abc407/clarifications) [🚀 Submit \(/contests/abc407/submit\)](/contests/abc407/submit)

[☰ Results ▾](#) [⬇️ Standings \(/contests/abc407/standings\)](/contests/abc407/standings)

[⬇️ Virtual Standings \(/contests/abc407/standings/virtual\)](/contests/abc407/standings/virtual)

[🔧 Custom Test \(/contests/abc407/custom_test\)](/contests/abc407/custom_test)

[📖 Editorial \(/contests/abc407/editorial\)](/contests/abc407/editorial)

[💬 Discuss \(https://codeforces.com/blog/entry/143155\)](https://codeforces.com/blog/entry/143155)



Official

F - Sums of Sliding Window Maximum (/contests/abc407/tasks/abc407_f) Editorial by en_translator (/users/en_translator)

Let us define the ordering of the elements of A by the lexicographical order of (A_i, i) . This way, the ordering of any pair of elements is uniquely determined.

For a (contiguous) subarray whose maximum value is A_i , let l be the number of elements to the left of A_i in the subarray, and r to the right. What condition should (l, r) satisfy?

Let L_i be the number of consecutive elements less than A_i to the left of A_i in A , and define R_i to the right likewise. (We will describe how to find these values afterward.)

Then, the subarray corresponding to (l, r) has A_i as the maximum value if and only if $0 \leq l \leq L_i$ and $0 \leq r \leq R_i$.

Here, write $x_{\min} := \min(l, r)$ and $x_{\max} := \max(l, r)$. Then the contribution of A_i to ans can be written as follows:

- For $0 < k \leq 1 + x_{\min}$, add $k \times A_i$ to $ans[k]$.
- For $1 + x_{\min} < k \leq 1 + x_{\max}$, add $(1 + x_{\min}) \times A_i$ to $ans[k]$.
- For $1 + x_{\max} < k \leq 2 + x_{\min} + x_{\max}$, add $(2 + x_{\min} + x_{\max} - k) \times A_i$ to $ans[k]$.

These are all affine function with a constant slope.

The cumulative sum trick is usually used to add a value to a specific subarray of a sequence, but it can be extended to affine functions as follows.

Let us replace a range addition as follows:

2025-05-24 (Sat)
23:47:07 -04:00

- Add $+A_i$ to $ans^{(2)}[1]$.
- Add $-A_i$ to $ans^{(2)}[2 + x_{min}]$.
- Add $-A_i$ to $ans^{(2)}[2 + x_{max}]$.
- Add $+A_i$ to $ans^{(2)}[2 + x_{min} + x_{max}]$.

After performing these operation for all ranges, repeat computing the cumulative sums twice, as follows:

- Let $ans^{(1)}[0] = ans^{(2)}[0]$ and $ans^{(1)}[i + 1] := ans^{(1)}[i] + ans^{(2)}[i + 1]$.
- $ans[0] = ans^{(1)}[0]$, $ans[i + 1] := ans[i] + ans^{(1)}[i + 1]$.

This way, we can find ans .

How to find L_i and R_i ?

Initialize the set of indices S by $\{0, N + 1\}$.

Inspect A_i in descending order of the value, and perform the following operation for each element:

1. Let L' be the maximum value less than i in S , and R' be the minimum value greater than i .
2. Let $L_i = i - L' - 1$ and $R_i = R' - i - 1$.
3. Insert i to S .

To perform step 1., we can use `std::set` in the standard library of C++, utilizing the `set.lower_bound` function.

The following is sample code in C++.

Copy

```

1. #include <iostream>
2. #include <string>
3. #include <vector>
4. #include <set>
5. #include <array>
6. using std::cin;
7. using std::cout;
8. using std::cerr;
9. using std::endl;
10. using std::string;
11. using std::pair;
12. using std::vector;
13. using std::set;
14. using std::min;
15. using std::max;
16. using std::array;
17. #include <atcoder/all>
18.
19.
20. typedef long long ll;
21. typedef pair<ll, ll> P;
22.
23. vector<ll> solve (const ll n, const vector<ll> &a) {
24.     vector<ll> ans(n+1, 0);
25.
26.     vector<P> asort(n);
27.     for (ll i = 0; i < n; i++) {
28.         asort[i] = {a[i], i};
29.     }
30.     sort(asort.begin(), asort.end());
31.
32.     array<vector<ll>, 3> integral;
33.
34.     integral[0].resize(n+3);
35.     for (ll i = 0; i <= n+2; i++) {
36.         integral[0][i] = 0;
37.     }
38.     set<ll> used = {-1, n};
39.     for (ll i = n-1; i >= 0; i--) {
40.         ll val = asort[i].first;
41.         ll idx = asort[i].second;
42.
43.         auto itgeq = used.lower_bound(idx);
44.         auto itle = itgeq; --itle;
45.
46.         // how many elements (on left/right) can a[idx] absorb as a maximum

```

m

2025-05-24 (Sat)
23:47:07 -04:00

```

47.         ll r = *itgeq - idx - 1;
48.         ll l = idx - *itle - 1;
49.
50.         ll xmin = min(l, r), xmax = max(l, r);
51.
52.         // ans[0 < i <= 1+min] <- i
53.         // ans[1+min < i <= 1+max] <- 1+min
54.         // ans[1+max < i <= 1+min+max] <- 1+min - (i - (1+max))
55.         // ans[1+min+max < i] <- 0
56.
57.         // ans'[0 < i <= 1+min] <- +1
58.         // ans'[1+min < i <= 1+max] <- 0
59.         // ans'[1+max < i <= 1+min+max] <- -1
60.         // ans'[1+min+max < i] <- 0
61.
62.         // ans''[1] <- +1
63.         // ans''[1+min+1] <- -1
64.         // ans''[1+max+1] <- -1
65.         // ans''[1+min+max+1] <- +1
66.         integral[0][1] += val * (+1);
67.         integral[0][1+xmin+1] += val * (-1);
68.         integral[0][1+xmax+1] += val * (-1);
69.         integral[0][1+xmin+xmax+2] += val * (+1);
70.
71.         used.insert(idx);
72.     }
73.     for (ll order = 1; order <= 2; order++) {
74.         integral[order].resize(n+3);
75.         integral[order][0] = 0;
76.         for (ll i = 1; i <= n+2; i++) {
77.             integral[order][i] = integral[order][i-1] + integral[order
-1][i];
78.         }
79.     }
80.
81.     for (ll i = 1; i <= n; i++) {
82.         ans[i] = integral[2][i];
83.     }
84.
85.
86.     return ans;
87. }
88.
89.
90. int main (void) {
91.     ll n;
92.     cin >> n;

```

2025-05-24 (Sat)
23:47:07 -04:00

```
93.     vector<ll> a(n);
94.     for (ll i = 0; i < n; i++) {
95.         cin >> a[i];
96.     }
97.     vector<ll> anslist = solve(n, a);
98.     for (ll i = 1; i <= n; i++) {
99.         cout << anslist[i] << "\n";
100.    }
101.
102.
103.     return 0;
104. }
```

posted: 44 minutes ago

last update: 44 minutes ago

am)

ps%3A%2F%2Fatcoder.jp%2Fcontests%2Fabc407%2Feditorial%2F13112%3Flang%3Den&title=Editorial%20-407)

[Rule \(/contests/abc407/rules\)](/contests/abc407/rules) [Glossary \(/contests/abc407/glossary\)](/contests/abc407/glossary)

[Terms of service \(/tos\)](/tos) [Privacy Policy \(/privacy\)](/privacy) [Information Protection Policy \(/personal\)](/personal) [Company \(/company\)](/company)
[FAQ \(/faq\)](/faq) [Contact \(/contact\)](/contact)

Copyright Since 2012 ©AtCoder Inc. (<http://atcoder.co.jp>) All rights reserved.

2025-05-24 (Sat)
23:47:07 -04:00