

7503. Разрушение дворцовой стены

 Решена

 Лёгкая

От Кодеруна всё ещё нет вестей, а проблемы становятся всё масштабнее.

На этот раз внезапно исчезли целые участки дворцовой стены. Вместо ровной кладки из блоков данных — зияющие дыры. Алгоритмы-строители уже на месте, но прежде чем приступить к восстановлению, им нужно понять, как выглядела стена до разрушения.

Стена состояла из n башен **разной** высоты. Высота i -й башни для $1 \leq i \leq n$ составляла ровно a_i метров. Согласно инженерным протоколам, необходимо рассмотреть все возможные непрерывные отрезки стены и для каждого из них вычислить вариативность по высоте — то есть разницу между самой высокой и самой низкой башней в этом отрезке. Более формально вариативность подотрезка определяется как: $\max|a_i - a_j|$ для $l \leq i, j \leq r$.

После этого все полученные значения складываются. Итоговая сумма и есть общая вариативность стены, которую необходимо найти.

Формат ввода

В качестве аргументов вашей функции передаются 2 параметра:

- Целое число n ($1 \leq n \leq 3 \cdot 10^5$) — общее количество башен.
- Одномерный массив целых чисел a размера n ($1 \leq a_i \leq 10^7$) — высоты башен. Гарантируется, что все числа a_i **различны**.

Формат вывода

В качестве ответа ваша функция должна вернуть одно число — общую вариативность стены. Гарантируется, что ответ помещается в 64-битовую целочисленную переменную.

Примечание

Первый тестовый пример

Существует всего один подотрезок для $l = r = 1$, его вариативность равна $|1 - 1| = 0$.

Второй тестовый пример

Существует три подотрезка:

- для подотрезка $l = r = 1$ вариативность равна 0;
- для подотрезка $l = r = 2$ вариативность тоже равна 0;
- для подотрезка $l = 1, r = 2$ вариативность равна $\max(|1 - 1|, |1 - 11|, |11 - 11|) = 10$.

Складывая все значения получаем, что вариативность стены равна 10.

Ограничения

Ограничение времени	2 с
Ограничение памяти	256 МБ

Пример 1

Ввод	Вывод
1 1	0

Пример 2

Ввод	Вывод
2 1 11	10




Пример 3

Ввод	Вывод
3 1 3 5	8

Теги

coderun boost challenge

C++ ▾

 Отправить

```
9  long long solve(int n, std::vector<long long> a) {
94     for (int i = n - 1; i > -1; i--) {
96         while (!R_border.empty()) {
102             };
103         };
104
105         if (!R_border.empty()) {
106             cnt_min[i] *= R_border.top() - i;
107         }
108         else {
109             cnt_min[i] *= n - i;
110         }
111         R_border.push(i);
112     };
113 };
114
115 //сумма максимумов
116 for (int i = 0; i < n; i++) {
117     count += cnt_max[i] * a[i];
118 }
119 //минус сумма минимумов
120 for (int i = 0; i < n; i++) {
121     count -= cnt_min[i] * a[i];
122 }
123 return count;
124 }
```