

Префиксные суммы. Разбор задач.

Шакир Бикметов

17 Марта 2024

1 Задача A

В данной задаче было достаточно посчитать префиксные суммы

$$\begin{aligned}pref_0 &= 0, \\pref_1 &= a_0, \\pref_2 &= a_0 + a_1, \\pref_3 &= a_0 + a_1 + a_2, \\&\dots, \\pref_n &= a_0 + a_1 + a_2 + \dots + a_{n-1}.\end{aligned}$$

Затем для ответа на каждый вопрос было достаточно посчитать сумму в отрезке от l до r ($a_l, a_{l+1}, a_{l+2}, \dots, a_{r-1}, a_r$) формулой $pref_{r+1} - pref_l$

```
1 int n;
2 vector<int> a;
3 // input n, a
4 // ...
5 vector<int> pref(n);
6 // calculate prefix sums
7 for (int i = 0; i < n; i++) {
8     pref[i + 1] = pref[i] + a[i];
9 }
10
11 int q;
12 cin >> q;
13 for (int i = 0; i < q; i++) {
14     int l, r;
15     cin >> l >> r;
16     // l, r are indices in array with first index 1
17     l--; r--; // this converts indices l, r to an array with first
                // index 0
18
19     // calculate answer
20     int ans = pref[r + 1] - pref[l];
21     // answer to the query
22     cout << ans << '\n';
23 }
```

Важно учитывать, что зачастую сумма элементов типа `int` не влезает в этот же тип, поэтому зачастую приходится использовать тип `long long` (64-битный знаковый целочисленный тип).

2 Задача В

Решение состоит из нескольких шагов:

- Научиться быстро считать сумму в произвольном прямоугольнике
- Перебрать все прямоугольники и взять максимум из их сумм

Шаг 1

Введем такое определение двумерной префиксной суммы:

$$pref_{i,j} = \sum_{k=0}^{i-1} \sum_{l=0}^{j-1} a_{k,l}$$

То есть $pref_{i,j}$ это сумма всех элементов прямоугольника с левым верхним углом в $(0, 0)$ и правым нижним углом в $(i-1, j-1)$, либо 0, если $i = 0$ или $j = 0$.

Аналогично одномерным префиксным суммам, можно вычислить формулу суммы элементов в прямоугольнике с левым верхним углом в (i_1, j_1) и правым нижним углом в (i_2, j_2) как

$$sum(i_1, j_1, i_2, j_2) = pref_{i_2+1, j_2+1} - (pref_{i_2+1, j_1} + pref_{i_1, j_2+1}) + pref_{i_1, j_1}$$

Используя эту же формулу можно вывести формулу для $pref_{i+1, j+1}$. Рассмотрим сумму прямоугольника размера 1 на 1 с элементом $a_{i,j}$:

$$a_{i,j} = sum(i, j, i, j) = pref_{i+1, j+1} - (pref_{i+1, j} + pref_{i, j+1}) + pref_{i, j}$$

Отсюда получаем:

$$\begin{aligned} pref_{i+1, j+1} &= sum(i, j, i, j) + (pref_{i+1, j} + pref_{i, j+1}) - pref_{i, j} \\ &= a_{i, j} + (pref_{i+1, j} + pref_{i, j+1}) - pref_{i, j} \end{aligned}$$

Шаг 2

У нас есть 4 координаты от 1 до 100, поэтому как оценку количества прямоугольников можно взять $100^4 = 10^8$. Учитывая, что вычисление суммы в прямоугольнике у нас за $O(1)$, то итогов получается порядка 10^8 действий, отсюда приблизительная оценка времени выполнения 0.5 секунд (из расчета $2 \cdot 10^8$ операций в секунду).

Перебрать координатый углов можно следующим образом:

```

1 for (int i1 = 0; i1 < n; i1++) {
2     for (int i2 = i1; i2 < n; i2++) {
3         for (int j1 = 0; j1 < m; j1++) {
4             for (int j2 = j1; j2 < m; j2++) {
5                 // logic
6             }
7         }
8     }
9 }

```

3 Задача С

Сопоставить одному типу учеников $+1$, другому -1 , затем выбор группы равносильно выбору отрезка с нулевой суммой.

Посчитаем префиксные суммы, тогда сумма в произвольном отрезке от l до r это $pref_{r+1} - pref_l$.

Зафиксируем правую границу r . Тогда чтобы сумма была нулю нужна такая граница l , чтобы $pref_{r+1} - pref_l = 0$ или $pref_l = pref_{r+1}$, то есть для количества искомым отрезков с заданной границей r нужно посчитать кол-во таких левых границ $l \leq r$, что $pref_l = pref_{r+1}$.

Пройдемся по всему массиву префикс-сумм слева направо и будем любым удобным способом поддерживать счетчик cnt_v , равный количеству границ $l \leq r$, что $pref_l = v$.

После каждой итерации добавляется 1 новая левая граница $l = i$, то есть нужно сделать $cnt[pref[i]] += 1$.

4 Задача D

Рассмотреть как изменения действуют на двумерный разностный массив – преобразование обратное двумерным префиксным суммам. Выйдет, что каждый запрос делает $O(1)$ изменений на разностный массив.

Храним разностный массив, выполняем запросы над разностным массивом, считаем в конце префиксную сумму и получаем ответ.

5 Задача E

Введем массив a размера $n - 1$:

$$a_i = \begin{cases} 1, & \text{if } s_i = s_{i+1}, \\ 0, & \text{if } s_i \neq s_{i+1}. \end{cases}$$

Тогда ответ на запрос – сумма в некотором отрезке массива a . Задача сводится к задаче А.

6 Задача F

Эта задача на другой линейный алгоритм – префиксный минимум. Префиксный минимум – это массив из минимумов каждого префикса.

Для i -ого дня выгоднее всего было купить мясо в день с минимальной стоимостью среди дней $j \leq i$, а это есть $(i + 1)$ -ый префиксный минимум.

Суммируем затраты для каждого дня и получаем ответ.

7 Задача G

В этой задаче будут только подсказки к решению:

Подсказка 1

Рассмотрите задачу, где вместо любых подпоследовательностей будут только подпоследовательности из подряд идущих элементов, то есть подотрезки.

Подсказка 2

Так как важна только кратность числу N , достаточно рассмотреть все числа по модулю N . Тогда кратность подотрезка равносильна тому, что сумма в подотрезке равна 0 по модулю N .

Подсказка 3

Перейти к префиксным суммам, расписать в ее терминах результат подсказки 2.