

SQRT декомпозиция

Задача 1.

Есть массив a из n элементов. Над ним выполняют операции:

- 1) $a[i] += x$
- 2) $\text{print } a[L] + \dots + a[R]$

Будем хранить два массива

- 1) собственно a
- 2) массив b длины \sqrt{n} , где $b[i] = a[i / \sqrt{n}] + \dots + a[(i + 1) / \sqrt{n} - 1]$

$a = \{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\}$

$b = \{6\ 15\ 24\}$

Как отвечать на запросы:

- 1) $a[i] += x$, $b[i / \sqrt{n}] += x$
- 2) суммируем $b[L / \sqrt{n} + 1] + \dots + b[R / \sqrt{n} - 1]$ и еще несколько элементов справа и слева. Осторожно, когда $L / \sqrt{n} + 1 > R / \sqrt{n} - 1$.

```
for (int i = L; i <= R; ) {
    // если начало блока и если все элементы блока попадают в запрос
    if (i % sqrtN == 0 && i + sqrtN - 1 <= R) {
        sum += b[i / sqrtN];
        i += sqrtN;
    } else {
        sum += a[i];
        i++;
    }
}
```

Медленно работает деление и модуль. Поэтому все значения i / \sqrt{n} можно сложить в массив заранее: $\text{block}[i] = i / \sqrt{n}$, потом смотреть в $b[\text{block}[i]]$.

Аналогично проверка $i \% \sqrt{n} == 0$: можно создать булевский массив $\text{isBegin}[i] = (i \% \sqrt{n} == 0)$

Задача 2.

Есть массив a из n элементов. Над ним выполняют операции:

- 1) $a[i] += x$
- 2) $\text{print } \max(a[L], \dots, a[R])$

Как отвечать на запросы:

- 1) $a[i] += x$, потом обновляем $b[i / \sqrt{n}]$. Если $x \geq 0$, можно просто сделать $b[i / \sqrt{n}] = \max(b[i / \sqrt{n}], a[i])$. А если $x < 0$, придется пробежаться по всему блоку и найти максимум на этом блоке заново.
- 2) То же самое, что и в случае с суммой

Задача 3.

Есть массив a из n элементов. Над ним выполняют операции:

- 1) $a[i] = x$ (присвоение)
- 2) $\text{print max}(a[L], \dots, a[R])$

Аналогично предыдущему случаю, т.к. $a[i] = x$ это то же самое, что $a[i] += x - a[i]$

Задача 4.

Есть массив a из n элементов. Над ним выполняют операции:

- 1) $a[L] += x, \dots, a[R] += x$ (прибавление x ко всему отрезку сразу)
- 2) $\text{print } a[L] + \dots + a[R]$

[0 0 0 0 0 0 0 0]
[0 0 0]

попытка хранить во втором массиве сумму

$\text{add}(1,7,5)$
[0 5 5 0 0 0 5 5 0]
[0 15 0] - не работает, $\text{ask}(5,8)$ пофейлится

попытка хранить во втором массиве, сколько прибавили к каждому элементу блока (не суммарно)

$\text{add}(1,7,5)$
[0 5 5 0 0 0 5 5 0]
[0 5 0]
 $\text{add}(5,8,2)$
[0 5 5 0 0 2 5 5 0]
[0 5 2]

$\text{ask}(3,8)$ - это два полных блока, и то что мы прибавили 2 к 5-ому элементу, мы никак не узнаем, т.к. полный блок мы обязаны обрабатывать смотря только во второй массив

правильное решение

Храним оба массива длиной \sqrt{n} :

$\text{sum}[k]$ - это сумма $a[k*\sqrt{n}] + \dots + a[(k+1)*\sqrt{n} - 1]$

$\text{add}[k]$ - это сколько прибавили к каждому из $a[k*\sqrt{n}], \dots, a[(k+1)*\sqrt{n} - 1]$.

Теперь просто i -ый элемент массива мы будем получать как $a[i] + \text{add}[i / \sqrt{n}]$

- 1) Проходимся for-ом, таким же, как выше, который использовался для ответа на запросы
 - a) Если блок полностью входит:
 - i) $\text{add}[i / \sqrt{n}] += x$
 - ii) $\text{sum}[i / \sqrt{n}] += \text{длина блока} * x$ (длина последнего блока может быть меньше \sqrt{n} , надо быть с этим внимательным)
 - b) Если не полностью:
 - i) $a[i] += x$

- ii) $\text{sum}[i / \text{sqrtN}] += x$
- 2) Теперь запросы на сумму. Точно так же проходимся форум.
- a) Если блок полностью входит:
 - i) $\text{result} += \text{sum}[i / \text{sqrtN}]$
- b) Если не полностью:
 - i) $\text{result} += a[i] + \text{add}[i / \text{sqrtN}]$

```
[0 0 0 0 0 0 0 0]
[0 5 5 5 5 5 5 0]
[0 5 5 5 5 7 7 2]
```

```
a = [0 0 0 0 0 0 0 0]
sum = [0 0 0]
add = [0 0 0]
>>>>>> add(1,7,5)
a = [0 5 5 0 0 0 5 5 0]
sum = [0 15 0]
add = [0 5 0]
>>>>>> add(5,8,2)
a = [0 5 5 0 0 2 5 5 0]
sum = [0 17 6]
add = [0 5 2]
>>>>>> ask(2,6) = 29
```

Задача 5.

Есть массив a из n элементов. Над ним выполняют операции:

- 1) $a[L] = x, \dots, a[R] = x$ (присвоение x ко всему отрезку сразу)
- 2) $\text{print } a[L] + \dots + a[R]$

Массивы, которые надо хранить:

- 1) $\text{sum}[k]$ - это сумма $a[k \cdot \text{sqrtN}] + \dots + a[(k+1) \cdot \text{sqrtN} - 1]$
- 2) $\text{value}[k]$ - это то, что щас присвоено элементам $a[k \cdot \text{sqrtN}], \dots, a[(k+1) \cdot \text{sqrtN} - 1]$
- 3) $\text{isSet}[k]$ - булевский массив - присвоено ли щас что-нибудь k -ому блоку

Ответы на запросы:

- 1) Запрос присвоения. Проходимся for-ом
 - a) Если блок полностью входит:
 - i) $\text{value}[i / \text{sqrtN}] = x$
 - ii) $\text{isSet}[i / \text{sqrtN}] = \text{true}$
 - iii) $\text{sum}[i / \text{sqrtN}] = \text{длина блока} * x$ (длина последнего блока может быть меньше sqrtN , надо быть с этим внимательным)
 - b) Если не полностью:

- i) возможно, стоит $\text{isSet}[i / \text{sqrtN}] = \text{true}$. Это значит, что актуальное значение $a[i]$ находится в $\text{value}[i / \text{sqrtN}]$, а не в $a[i]$. Придется пропустить значение $\text{value}[i / \text{sqrtN}]$ всем элементам в (i/sqrtN) -ом блоке.
 - ii) Пройдемся форум по этому блоку и поставим $a[i] = \text{value}[i / \text{sqrtN}]$.
 - iii) После этого проставим $\text{isSet}[i / \text{sqrtN}] = \text{false}$. Теперь последующие итерации по циклу i не будут триггерить пункты (i, ii). А все элементы (i/sqrtN) -ого блока имеют актуальные значения.
 - iv) $\text{sum}[i / \text{sqrtN}] += x - a[i]$
 - v) $a[i] = x$
- 2) Теперь запросы на сумму. Точно так же проходимся форум.
- a) Если блок полностью входит:
 - i) $\text{result} += \text{sum}[i / \text{sqrtN}]$
 - b) Если не полностью:
 - i) $\text{result} += \text{isSet}[i / \text{sqrtN}] ? \text{value}[i / \text{sqrtN}] : a[i]$

Задачи (еще будут добавлены)

Осторожно, некоторые трудные

<http://acm.timus.ru/problem.aspx?space=1&num=1061>

<http://acm.timus.ru/problem.aspx?space=1&num=1126>

<http://acm.timus.ru/problem.aspx?space=1&num=1316>

<http://acm.timus.ru/problem.aspx?space=1&num=1491>

<http://acm.timus.ru/problem.aspx?space=1&num=1890>

<http://acm.timus.ru/problem.aspx?space=1&num=2014>

<http://codeforces.com/problemset/problem/56/E>

<http://codeforces.com/problemset/problem/103/D>