

Дерево Фенвика

Илья Булгаков, Илья Гусев

Московский физико-технический институт

Москва, 2023

Содержание

1 Задача: вычисление суммы на подотрезке

2 Дерево Фенвика

- Побитовое И/Или
- Массив сумм
- Дерево Фенвика

Задача: вычисление суммы на подотрезке и модификация в точке

Дан массив $a_0, a_1 \dots, a_n$.

Запрос 1: замена числа на i -ой позиции на x : $a_i = x$

Запрос 2: сумма на отрезке $[l, r]$: $\sum_{i=l}^r a_i$

Вспоминаем

Как мы уже умеем решать эту задачу?

Побитовое И/Или

Вспоминаем

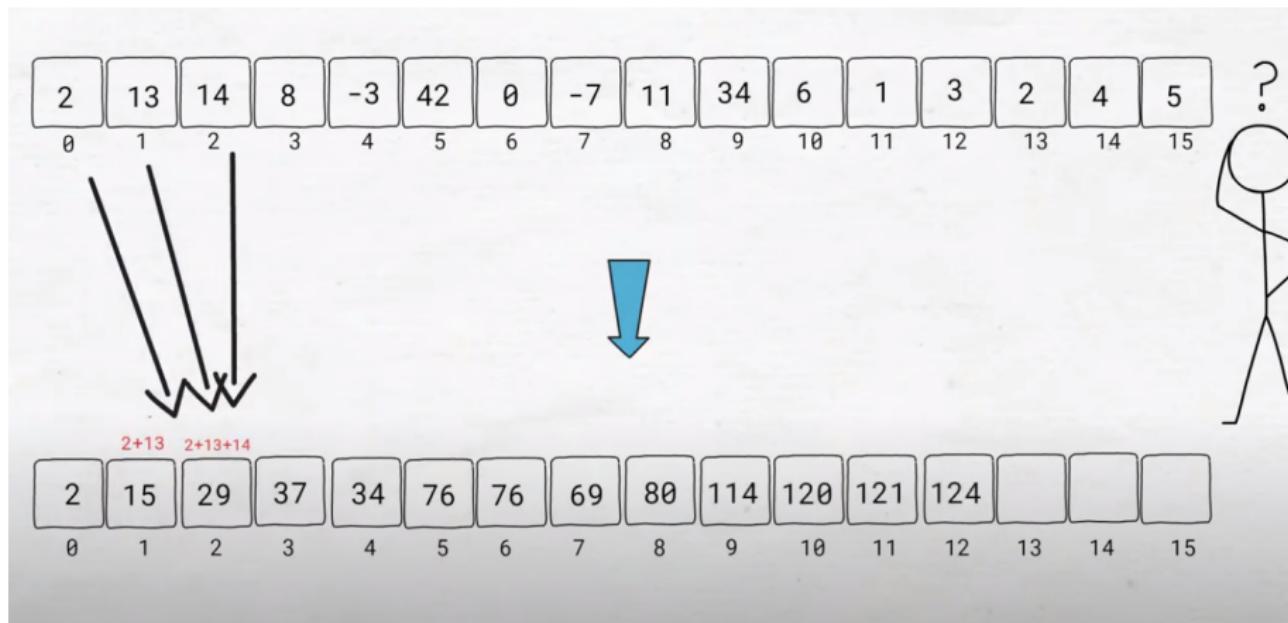
Что за операции? Вспомним представление чисел.

ИЛИ	0011
0101	
	0111

И	0011
	0101
	0001

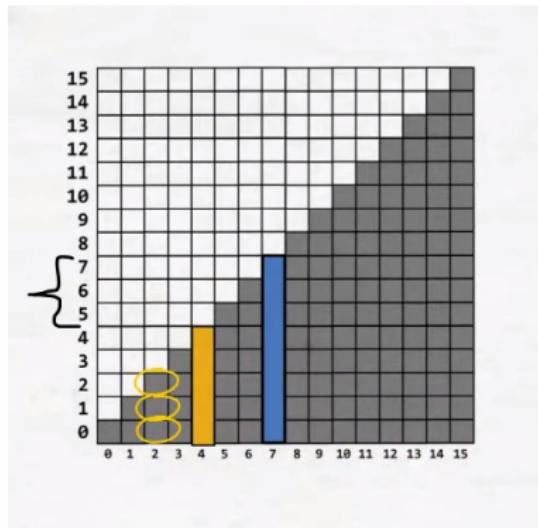
Массив сумм

Строим массив, который в i -ой ячейке содержит сумму элементов исходного массива от 0-го до i -го



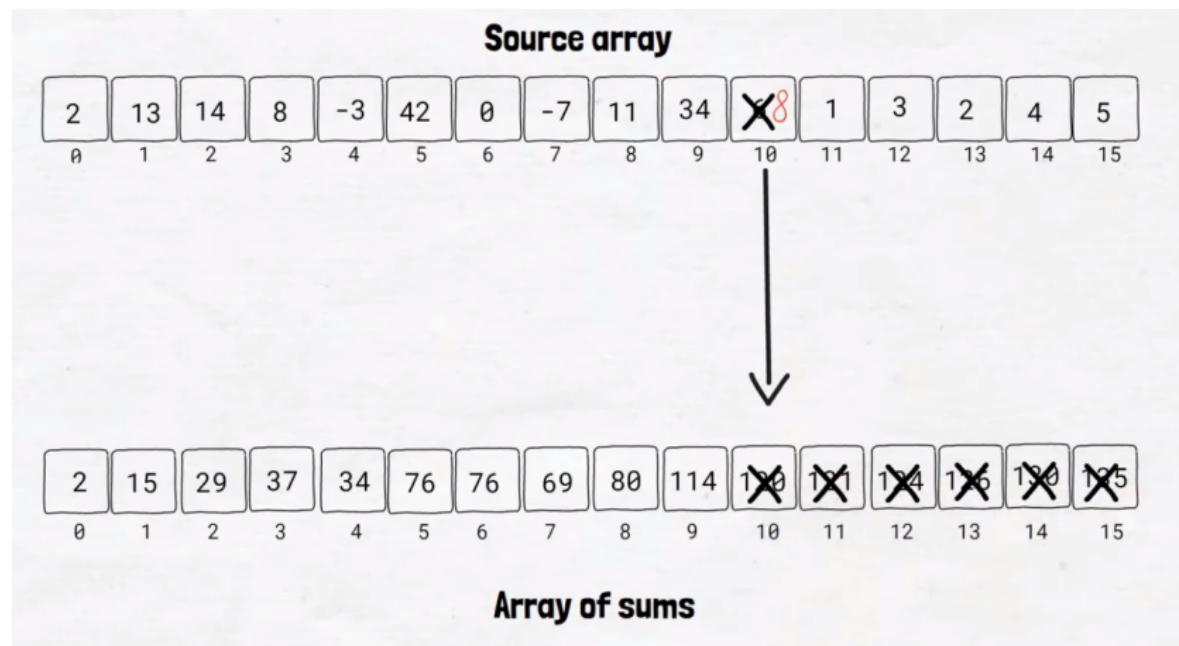
Массив сумм: вычисление суммы на подотрезке

Вычисляется очень удобно: разница двух сумм. Чтобы вычислить сумму элементов на отрезке [5;7], т.е. с 5го по 7ой включительно, то надо из значения по 7-му индексу вычесть значение по 4ому индексу.



Массив сумм: модификация

Проблема: после модификации i -го элемента требуется пересчитать все суммы, идущие от i .



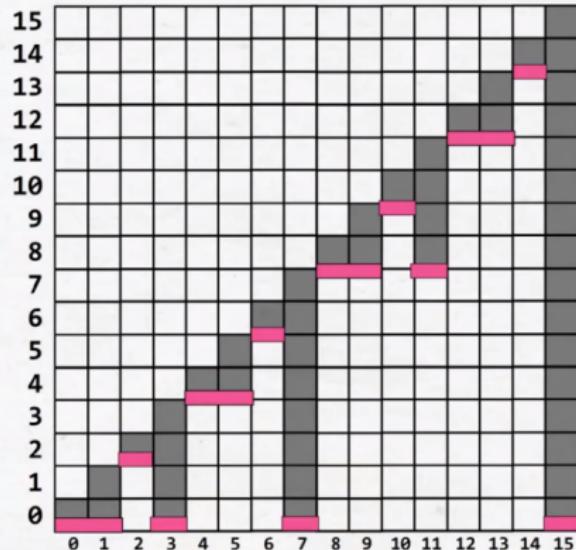
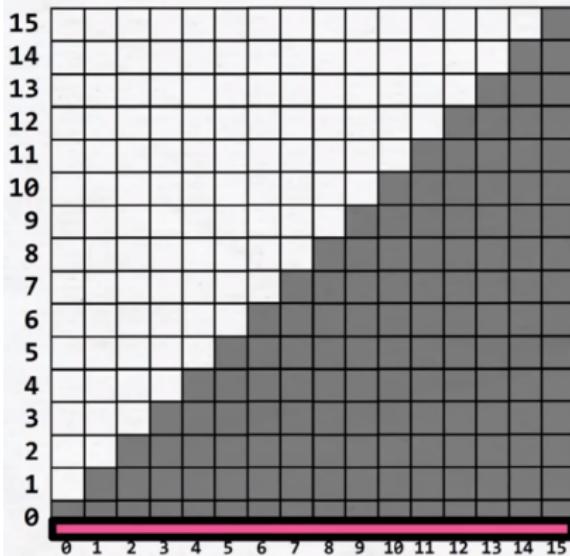
Сравнение алгоритмов

- ① Работа на простом массиве
- ② Работа на массиве сумм
- ③ Дерево Фенвика

	Update element	Get sum
Source array	$O(1)$	$O(n)$
Array of sums	$O(n)$	$O(1)$
Fenwick tree	$O(\log(n))$	$O(\log(n))$

Дерево Фенвики: идея

Вычислим заранее суммы, но не просто с 0го до i -ый, а более хитро: от какого-то до i -го.



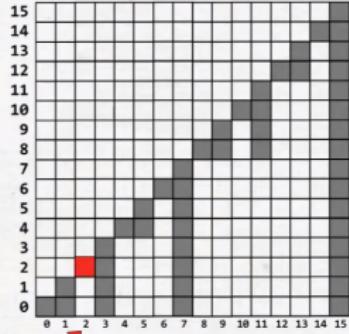
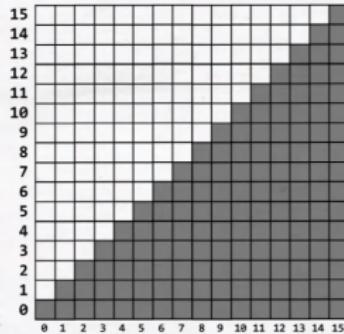
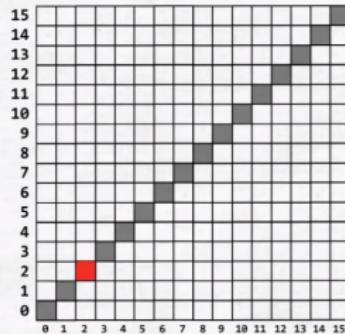
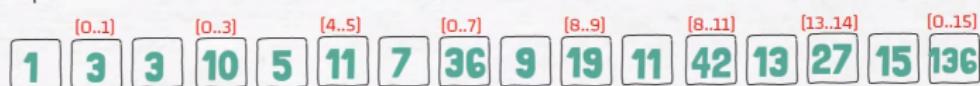
Дерево Фенвика: идея

На нечетных будет стоять сам элемент

Source array



Fenwick tree



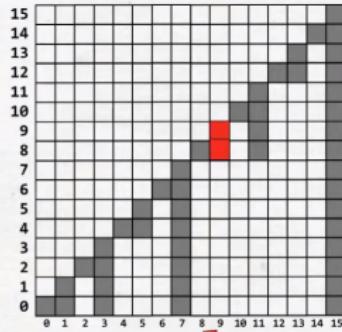
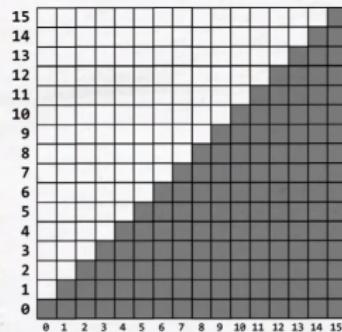
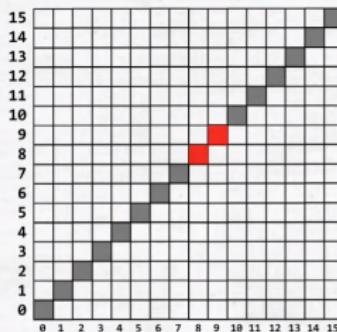
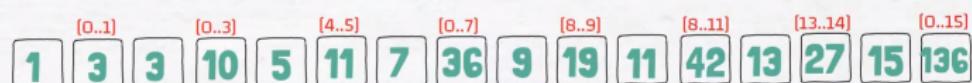
Дерево Фенвика: идея

На четных сумма от какого-то элемента до текущего

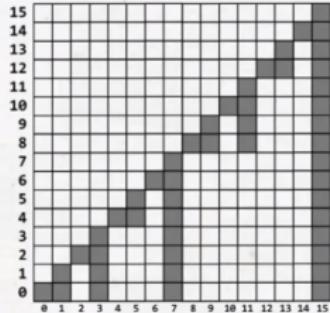
Source array



Fenwick tree



Дерево Фенвика: идея



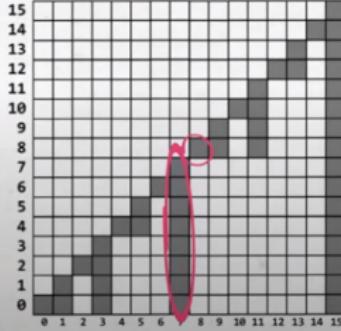
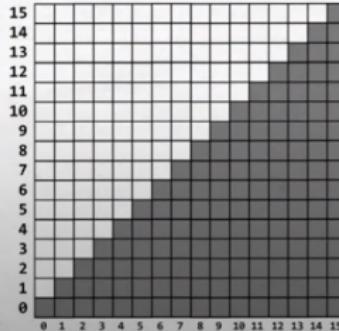
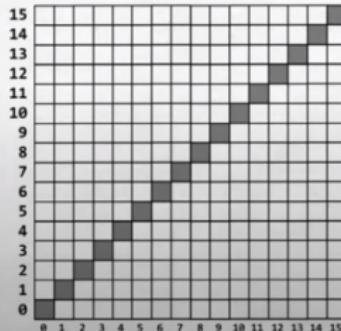
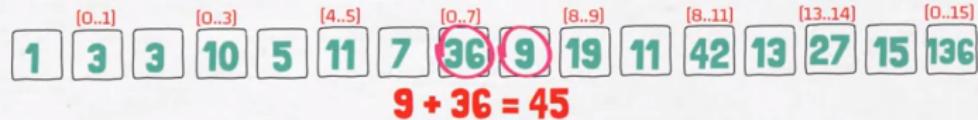
Дерево Фенвика: идея

Как найти сумму от 0 элемента до 8? Через сумму двух элементов на предподсчете.

Source array



Fenwick tree

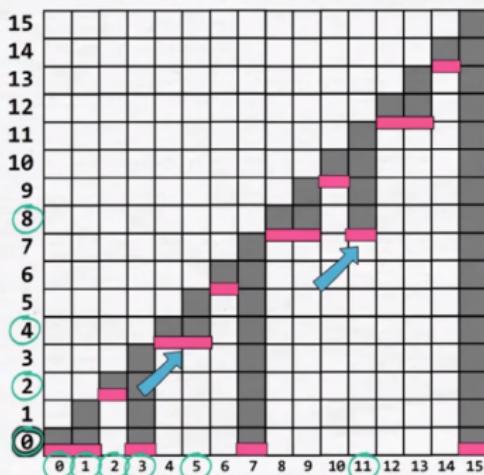


Дерево Фенвики: идея

Как найти сумму на произвольном отрезке? По той же методике - через разность двух сумм, $\text{sum}(6,8) = \text{sum}(0,8) - \text{sum}(0,5)$

Дерево Фенвики: формула для вычисления нижней границы

По этой формуле можно вычислить нижнюю границу отрезка

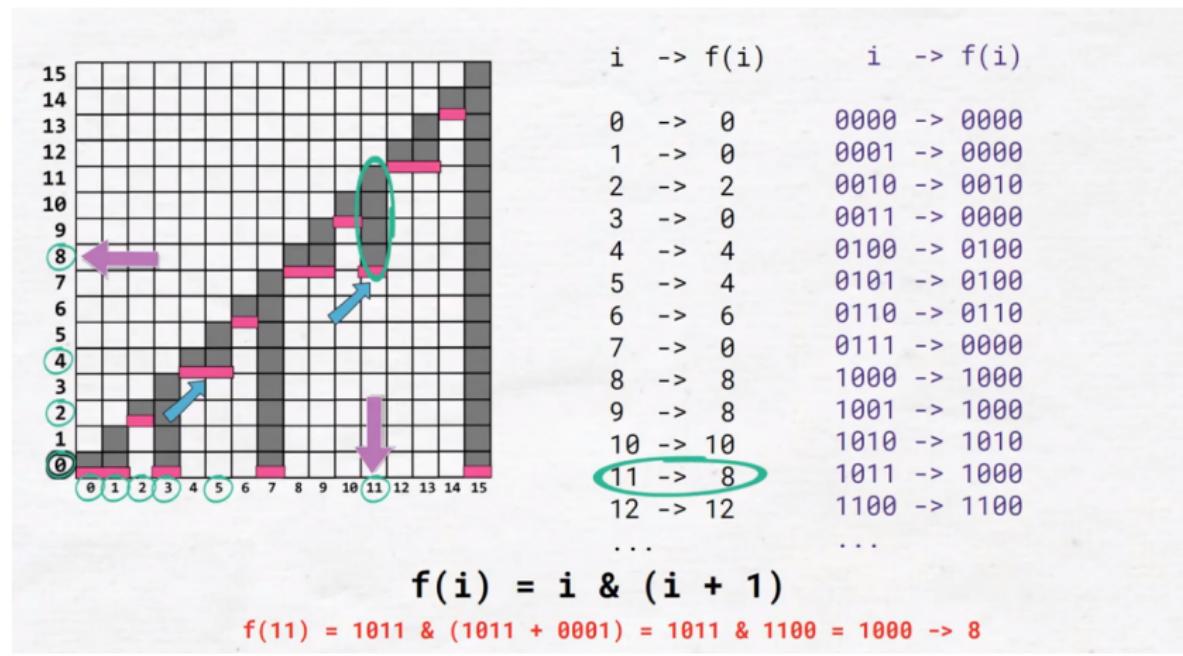


$i \rightarrow f(i)$	$i \rightarrow f(i)$
0 -> 0	0000 -> 0000
1 -> 0	0001 -> 0000
2 -> 2	0010 -> 0010
3 -> 0	0011 -> 0000
4 -> 4	0100 -> 0100
5 -> 4	0101 -> 0100
6 -> 6	0110 -> 0110
7 -> 0	0111 -> 0000
8 -> 8	1000 -> 1000
9 -> 8	1001 -> 1000
10 -> 10	1010 -> 1010
11 -> 8	1011 -> 1000
12 -> 12	1100 -> 1100
...	...

$$f(i) = i \& (i + 1)$$

Дерево Фенвики: формула для вычисления нижней границы

Пример



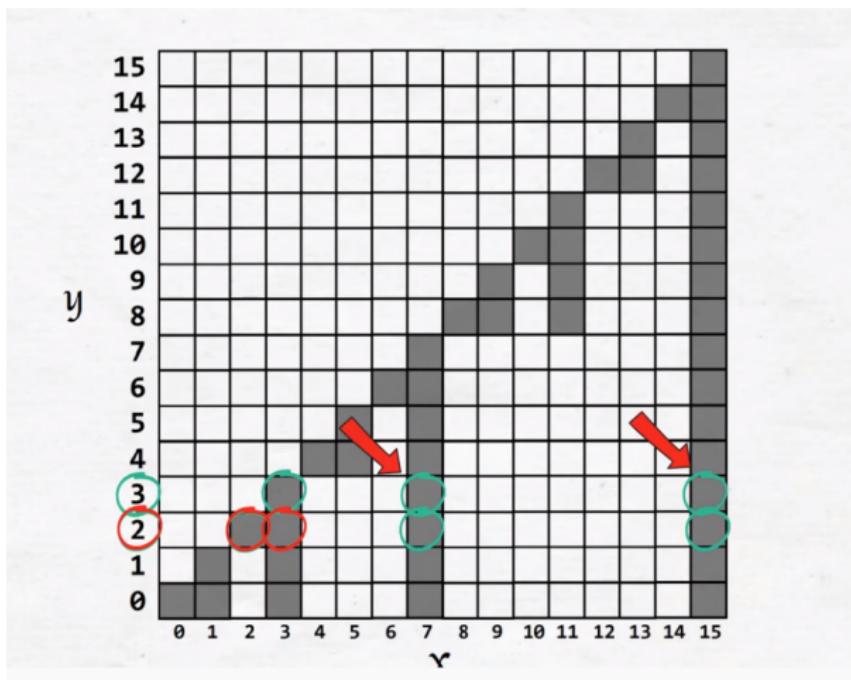
Дерево Фенвика: формула для подсчёта суммы на отрезке

$$\textcircled{1} \quad S_{l:r} = S_r - S_{l-1}$$

$$\textcircled{2} \quad S_i = S_i + S_{f(i)-1} + S_{f(f(i)-1)-1} + \dots$$

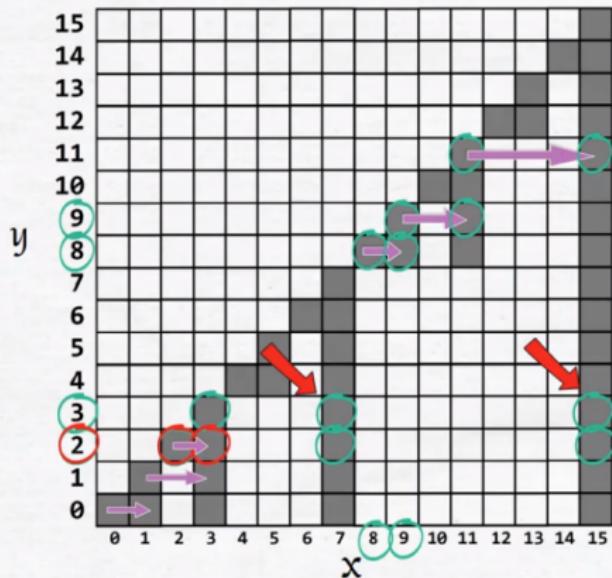
Дерево Фенвики: модификация элемента

Посмотрим, какие суммы нам надо модифицировать, если поменяется элемент 2 или 3?



Дерево Фенвики: модификация элемента

Формула

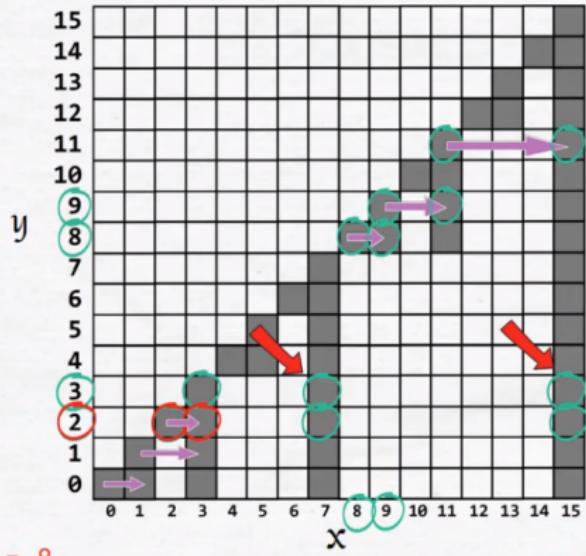


$$f(i) = i \mid (i + 1)$$

$i \rightarrow f(i)$	$i \rightarrow f(i)$
0 -> 1	0000 -> 0001
1 -> 3	0001 -> 0011
2 -> 3	0010 -> 0011
3 -> 7	0011 -> 0111
4 -> 5	0100 -> 0101
5 -> 7	0101 -> 0111
6 -> 7	0110 -> 0111
7 -> 15	0111 -> 1111
8 -> 9	1000 -> 1001
9 -> 10	1001 -> 1010
10 -> 11	1010 -> 1011
11 -> 15	1011 -> 1111
12 -> 13	1100 -> 1101
13 -> 15	1101 -> 1111
14 -> 15	1110 -> 1111
15 -> ...	

Дерево Фенвики: модификация элемента

Пример: модифицируем элемент



$$f(8) = 8 | (1 + 8) \rightarrow 1000 | 1001 = 1001 \rightarrow 9$$

$$f(9) = 9 | (1 + 9) \rightarrow 1001 | 1010 = 1011 \rightarrow 11$$

$i \rightarrow f(i)$	$i \rightarrow f(i)$
0 -> 1	0000 -> 0001
1 -> 3	0001 -> 0011
2 -> 3	0010 -> 0011
3 -> 7	0011 -> 0111
4 -> 5	0100 -> 0101
5 -> 7	0101 -> 0111
6 -> 7	0110 -> 0111
7 -> 15	0111 -> 1111
8 -> 9	1000 -> 1001
9 -> 10	1001 -> 1010
10 -> 11	1010 -> 1011
11 -> 15	1011 -> 1111
12 -> 13	1100 -> 1101
13 -> 15	1101 -> 1111
14 -> 15	1110 -> 1111
15 -> ...	

Дерево Фенвики: свойства

0-индексация. Для 1-индексации немного другие функции.

$f(i) = i \& (i + 1)$ - левые границы отрезка с правой границей в i

$g(i) = i | (i + 1)$ - индексы для обновления

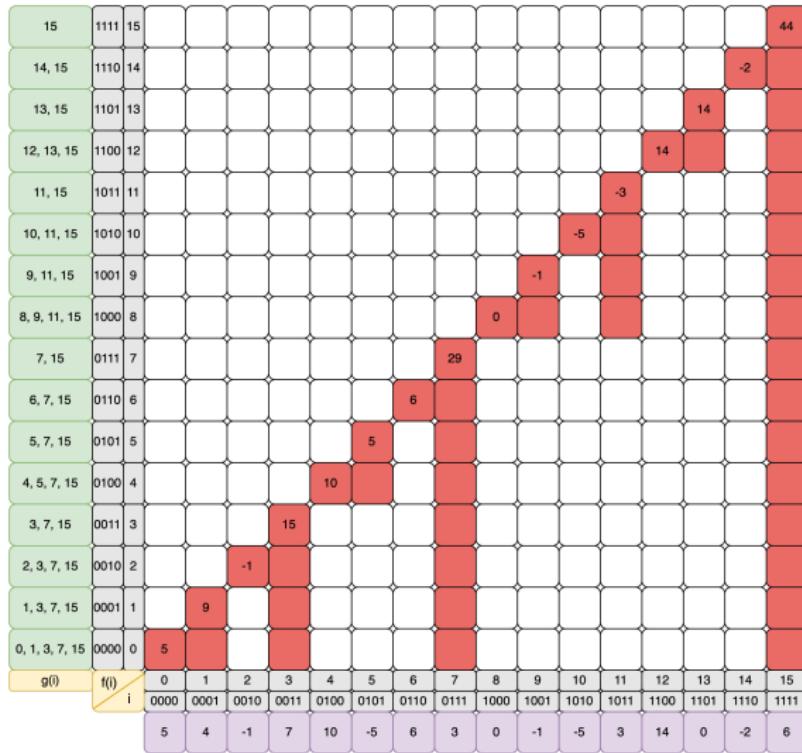
Свойства $f(i)$:

- ❶ Зануление всех подряд идущих единиц в конце двоичной записи числа
- ❷ Для всех чётных i : $f(i) = i$
- ❸ Для всех нечётных i : $f(i) < i$
- ❹ Для всех $i = 2^j - 1$: $f(i) = 0$

Свойства $g(i)$:

- ❶ Замена последнего нуля на 1
- ❷ Для всех чётных i : $g(i) = i + 1$
- ❸ Для всех нечётных i : $g(i) > i + 1$
- ❹ Для всех $i = 2^j - 1$: $g(i) = 2^{j+1} - 1$

Отрезки в дереве Фенвики



Реализация дерева Фенвики

Особенности

- Оформлено как класс
- Используем битовые операции в C++
- Проговорить интерфейс класса

Дерево Фенвики

Пример.

Найти количество элементов массива $\{a_1, \dots, a_n\}$ в подотрезке $\{a_l, \dots, a_r\}$, больших заданного k .

Дерево Фенвики

```
struct node {
    int pos; // query index
    int val; //
    int l; // left border
    int r; // right border
};

bool comp(node a, node b)
{
    if (a.val == b.val)
        return a.l > b.l;

    return a.val > b.val;
}
```

Дерево Фенвики

$$n_{idx+} = 1, n_{g(idx)+} = 1, \dots$$

```
void modify(std::vector<int> BIT, int n, int idx)
{
    while (idx < n) {
        BIT[idx]++;
        idx = idx | (idx + 1);
    }
}
```

$$N_{idx} = n_{idx} + n_{f(idx)-1} + \dots$$

```
int query(std::vector<int> BIT, int idx)
{
    int ans = 0;
    while (idx) {
        ans += BIT[idx];
        idx = idx & (idx + 1);
    }
    ans += BIT[0];
    return ans;
}
```

Дерево Фенвика

```
std::cin >> n >> q;
for (int i = 0; i < n; ++i)
    std::cin >> a[i];
nodes[i] = {0, a[i], 0, i};

for (int i = n; i < n + q; ++i)
    std::cin >> k >> l >> r;
nodes[i] = {i - n + 1, k, l, r};

sort(nodes, nodes + n + q, comp);

std::vector<int> BIT(n);
std::vector<int> ans(q);

for (int i = 0; i < n + q; ++i) {
    if (nodes[i].pos != 0) {
        /*N_lr = N_r - N_{l-1}*/
        cnt = query(BIT, nodes[i].r) - query(BIT, nodes[i].l - 1);
        ans[nodes[i].pos] = cnt;
    }
    else
        update(BIT, n, nodes[i].r);
}
```

Дерево Фенвики

Задача.

К массиву неотрицательных чисел a_1, a_2, \dots, a_n поступает q запросов двух типов:

- ① изменить значение элемента $a_{pos} = x$;
- ② по числу k сообщить минимальное r , такое что $a_1 + a_2 + \dots + a_r \geq k$

Обработайте запросы за $O((n + q)\log n)$

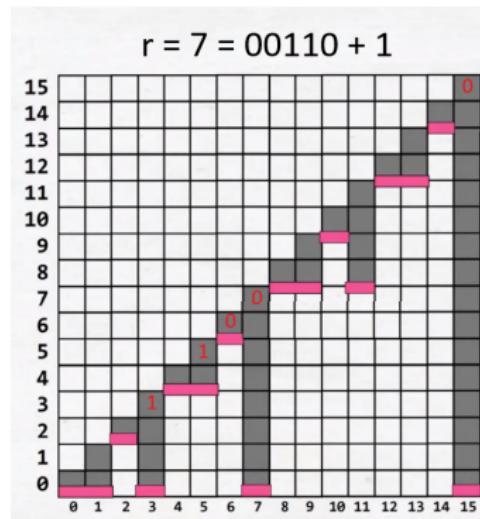
Дерево Фенвики

Решение.

Построим дерево Фенвики с префиксными суммами $s(i)$. Пусть

$$h(i) = \begin{cases} f(i) & s(i) \geq k \\ g(i) & s(i) < k \end{cases}, b(i) = \begin{cases} 1 & s(i) \geq k \\ 0 & s(i) < k \end{cases}$$

$$r = \sum_{j=0}^{\lfloor \log(n) \rfloor} b(h(a_{j-1}^*)) * 2^{\lfloor \log(n) \rfloor - j} + 1, a_{-1}^* = 2^{\lfloor \log(n) \rfloor}, a_j^* = h(a_{j-1}^*)$$



Полезные ссылки |

-  Оригинальная статья Фенвика
<https://shorturl.at/erACS>
-  Neerc: Дерево Фенвика
<https://shorturl.at/vSTW2>
-  youtube: Дерево Фенвика
<https://www.youtube.com/watch?v=muW1t0yqUZ4>