Дерево отрезков.

Дерево отрезков - структура данных, которая позволяет быстро отвечать на следующие запросы:

- Найти сумму (минимум) всех значений массива на отрезке $[l;r]$

- Сделать значение i-того элемента массива равным a.

Дерево отрезков позволяет обрабатывать оба запроса за сложность $O(log(n))$, где n - количество элементов в массиве. Деревот орезков - это бинарное дерево, в котором каждая вершина задает некоторый отрезок, а ее дети - левую и правую половину данного отрезка.

Разберем подробней случай минимума. Пусть m - минимальное число, такое что $m >= n$ и m является степенью двойки. Дозаполним наш исходный массив A до m элементов значениями inf. Заведем массив t, отвечающий за все вершины дерева отрезков. Пусть $t[1]$ - вершина, которая отвечает за весь массив A (то есть в ней будет храниться минимум по всем элементам массива). Сгенерируем наше дерево таким образом, что детьми вершины $t[i]$ будут являться вершины $t[i*2]$ и $t[i*2+1]$. Тогда для первой вершины детьми будут вершины 2 и 3. Вершина $t[2]$ будет хранить минимум на отрезке $[0; m/2-1]$, $t[3]$ - минимум на отрезке $[m/2; m-1]$. Таким образом за отреки длины 1 будут отвечать все вершины дерева $t[i]$, где $i >= m$. То есть $t[i+m] = A[i]$ для любого элемента массива A. Всего в дереве $2*m-1$ элементов (однако, есть еще элемент с индексом 0, который не является вершиной дерева и не содержит в себе никакой информации).

Удобно завести структуру *stree*, в которой будет храниться вся информация по дереву отрезков:

```
struct stree
{
    int inf;
    int n;
    vector<int> tmin;
    void build (const vector<int> &A)
    { ......... }
    void set (int i, int x)
    { ......... }
    int getmin (int l, int r)
    { ........ }
};
```

Приведем код построения дерева из исходного массива A:

```
void build (const vector<int> &A)
{
    inf = 2000000000;
    n = 1;
    while (n < A.size())
        n *= 2;
    tmin.assign(2 * n, inf);
```

```
    for (int i = 0; i < A.size(); ++i)
        tmin[i + n] = A[i];
    for (int i = n - 1; i >= 1; --i)
        tmin[i] = min(tmin[i * 2],tmin[i * 2 + 1]);
}
```

Теперь приведем код изменения значения одного элемента в массиве:

```
void set (int i, int x)
{
    i = i + n;
    tmin[i] = x;
    while (true)
    {
        i /= 2;
        if (i == 0)
            break;
        tmin[i] = min(tmin[i * 2],tmin[i * 2 + 1]);
    }
}
```

И код запроса минимума на отрезке $[l; r]$:

```
int getmin (int l, int r)
{
    l = l + n;
    r = r + n;
    int resmin = inf;
    for (; l <= r; l/= 2, r /= 2)
    {
        if (l % 2 == 1)
            resmin = min(resmin, tmin[l++]);
        if (r % 2 == 0)
            resmin = min(resmin, tmin[r--]);
    }
    return resmin;
}
```