```cpp
#include "../../bits/stdc++.h"
// 非再帰セグ木 1-indexed
// http://d.hatena.ne.jp/komiyam/20131202/1385992406
// http://smijake3.hatenablog.com/entry/2018/11/03/100133
// iの親: i/2
// iの子: i*2, i*2+1
// iの兄弟: i^1
// iの深さ: log_2(i)
// iの区間幅: N/highest(i) (highest(i):=iを超えない最大の2冪)
// iの区間左端: (i-highest(i))*width(i)
// 左からx番目の葉
// RangeSumQuery: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3433275#1
// RangeMinimumQuery: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3433284#1
template <typename M>
class SegmentTree
{
    using T = typename M::type;
    int n;
    std::vector<T> node;

  public:
    // v を基に初期化
    SegmentTree(const std::vector<T> &v)
    {
        // n は v.size() 以上の最小の2冪
        n = 1;
        while (n < int(v.size()))
            n *= 2;
        node.resize(2 * n, M::id());

        // i の子 -> 2*i+1, 2*i+2 , 親 -> (i-1)/2
        for (int i = 0; i < int(v.size()); i++)
            node[i + n] = v[i];
        for (int i = n - 1; i >= 0; i--)
            node[i] = M::op(node[2 * i], node[2 * i + 1]);
    }
    // Monoid::id 初期化
    SegmentTree(int _n)
    {
        // n は v.size() 以上の最小の2冪
        n = 1;
        while (n < _n)
            n *= 2;
        node.resize(2 * n, M::id());
    }
    // x 番目を val に更新
    void update(int x, T val)
    {
        x += n;
        node[x] = val;
        while (x >>= 1)
        {
            node[x] = M::op(node[2 * x], node[2 * x + 1]);
        }
    }
    // v[x] を M::op(v[x], val) に更新
    void operation(int x, T val)
    {
        x += n;
        node[x] = M::op(node[x], val);
        while (x >>= 1)
        {
            node[x] = M::op(node[2 * x], node[2 * x + 1]);
        }
    }
    // [a, b) の op
    T query(int l, int r)
    {
        l += n;
        r += n;
        T retl = M::id(), retr = M::id();
        while (l < r)
        {
            if (l & 1)
                retl = M::op(retl, node[l++]);
            if (r & 1)
                retr = M::op(node[--r], retr);
            l >>= 1;
            r >>= 1;
        }
        return M::op(retl, retr);
    }
};

```