```cpp
#include "../bits/stdc++.h"
/**
 *  Sparse Table
 *  静的なデータ列の区間に対し，結合則/冪等性が成立する演算 op を
 *  構築 O(NlogN)，取得 O(1)
 */
// verified: https://www.spoj.com/problems/RMQSQ/ submission:23714430
struct RangeMinimumQuery
{
    using type = int;
    static type op(const type &l, const type &r) { return std::min(l, r); }
};

template <typename M>
class SparseTable
{
    using T = typename M::type;
    std::vector<std::vector<T>> memo;
    std::vector<int> clz;

  public:
    // v を基に構築
    SparseTable(const std::vector<T> &v)
    {
        int b = 0;
        while ((1 << b) <= static_cast<int>(v.size()))
            b++;
        memo.assign(b, std::vector<T>(1 << b));
        for (int i = 0; i < static_cast<int>(v.size()); i++)
            memo[0][i] = v[i];
        for (int i = 1; i < b; i++)
        {
            for (int j = 0; j + (1 << i) <= (1 << b); j++)
            {
                memo[i][j] = M::op(memo[i - 1][j], memo[i - 1][j + (1 << (i - 1))]);
            }
        }
        clz.assign(static_cast<int>(v.size()) + 1);
        for (int i = 2; i < static_cast<int>(clz.size()); i++)
        {
            clz[i] = clz[i >> 1] + 1;
        }
    }

    // [a, b) の op 結果
    T query(int l, int r)
    {
        int b = clz[r - l];
        return M::op(memo[b][l], memo[b][r - (1 << b)]);
    }
};

```