```cpp
/*
    | Segment Tree |
    Desc: Classic segment tree. Point Update Range Queries in O(n*log(n)).
    Source: KawakiMeido
    State: Untested lmao
*/

struct SegmentTree{
    struct Node{
        int val;
        Node(){
            val = INF;
        }
    };

    int n;
    vector<Node> IT;

    Node comb(Node l, Node r){
        if (l.val == INF) return r;
        if (r.val == INF) return l;

        Node i;
        //Update logic

        return i;
    }

    void build(int idx, int l, int r){
        if (l==r){
            //Update logic
            return;
        }

        int mid = (l+r)/2;
        build(idx*2,l,mid);
        build(idx*2+1,mid+1,r);
        IT[idx] = comb(IT[idx*2],IT[idx*2+1]);
    }

    void update(int idx, int l, int r, int x, Node val){
        if (r < x || x < l) return;
        if (l==r){
            //Update logic
            return;
        }
        int mid = (l+r)/2;
        update(idx*2,l,mid,x,val);
        update(idx*2+1,mid+1,r,x,val);
```

```
50          IT[idx] = comb(IT[idx*2],IT[idx*2+1]);
51      }
52
53      Node getNode(int idx, int l, int r, int x, int y){
54          if (y < l || r < x) return Node();
55          if (x <= l && r <= y){
56              return IT[idx];
57          }
58
59          int mid = (l+r)/2;
60          return comb(getNode(idx*2,l,mid,x,y),getNode(idx*2+1,mid+1,r,x,y));
61
62      }
63
64      void init(int _n){
65          n = _n;
66          IT.resize(n*4+10, Node());
67          build(1,1,n);
68      }
69  };
```