Merge Sort Segment Tree:

```
//find the smallest number greater or equal to a specified number
vector<vector<int>> seg(400100);
vector<int> a(100100);

vector<int> build(int p , int l , int r){
    if(l==r) return seg[p] = vector<int> (1 , a[l]);
    auto first = build(left);
    auto second = build(right);
    merge(first.begin() , first.end() , second.begin() , second.end() , back_inserter(seg[p]));
    return seg[p];
}

int query (int i , int j , int val , int p , int l , int r){
    if(j<l  ||  r<i) return inf; //to change
    if(i<=l && r<=j){
        auto pos = lower_bound(seg[p].begin() , seg[p].end() , val);
        if(pos!=seg[p].end()) return *pos;
        return inf;
    }
    return min(query(i , j , val , left) , query(i , j , val , right));
}

//to make point update query use multiset
vector<multiset<int>> seg(400100);

void update(int i , int val , int p , int l , int r){
    seg[p].erase(seg[p].find(a[i]));
    seg[p].insert(val);
    if(l!=r) {
        if(i <= (l+r)>>1)
            update(i , val , left);
        else
            update(i , val , right);
    }
    else a[i] = val;
}
```