```
//searching for the first element greater than <a> in a given range
int query(int i , int j , int k , int p , int l , int r){
    // check out of the range
    if(j<l  ||  r<i) return -1;

    // check inside the range
    if(i<=l && r<=j){
        // check not found case
        if(seg[p] <= k) return -1;

        // return the answer
        if(l==r) return l;

        // check the condition and go left or right
        if(seg[p<<1] > k){
            return query(i , j , k , left);
        }
        return query(i , j , k , right);
    }

    // go left and right to get the ans
    int ans = query(i , j , k , left);
    if(ans!=-1) return ans;
    return query(i , j , k , right);
}
```

Finding subsegments with the maximal sum:

```
// out of range case in query is node(0)
struct node {
    int sum , prefix , suffix , ans;

    node(){}

    node(int val){
        this->sum = val;
        this->prefix = this->suffix = this->ans = max(0, val);
    }
};

node merge(node l, node r) {
    node res;
    res.sum = l.sum + r.sum;
    res.prefix = max(l.prefix , l.sum + r.prefix);
    res.suffix = max(r.suffix , r.sum + l.suffix);
    res.ans = max({l.ans , r.ans , l.suffix+r.prefix});
    return res;
}
```