```
/* GSS1 - Can you answer these queries I
You are given a sequence A[1], A[2],...,A[N]. (|A[i]|≤15007, 1≤N≤50000).
A query is defined as follows:
Query(x,y) = Max { a[i]+a[i+1]+...+a[j] ; x ≤ i ≤ j ≤ y }.
Given M queries, your program must output the results of these queries.
*/
#include<bits/stdc++.h>
using namespace std;
#define ll long long
#define MAXN 50005
#define lf nd<<1
#define rg (nd<<1)+1
#define m (int)((b+e)>>1)
#define N tree[nd]
#define L tree[lf]
#define R tree[rg]

int a[MAXN];
struct data{
    ll prefix,suffix,bestSum,sum;
}tree[4*MAXN];

data maxData(data x, data y){
    data z;
    z.prefix = max(x.prefix, x.sum + y.prefix);
    z.suffix = max(y.suffix, y.sum + x.suffix);
    z.bestSum = max(max(x.bestSum, y.bestSum), x.suffix + y.prefix);
    z.sum    = x.sum + y.sum;
    return z;
}
void build(int nd,int b,int e){
    if(b==e){ N.prefix = N.suffix = N.bestSum = N.sum = a[b]; return; }
    build(lf,b,m);
    build(rg,m+1,e);
    N = maxData(L,R);
}
data query(int nd,int b,int e,int x,int y){
    if(b>y || e<x){
        data INF;
        INF.prefix = INF.suffix = INF.bestSum = INF.sum = -1000000000;
        return INF;
    }
    if(b>=x && e<=y) return tree[nd];
    data f1 = query(lf,b,m,x,y);
    data f2 = query(rg,m+1,e,x,y);
    return maxData(f1,f2);
}
int main(){
    int n; scanf("%d",&n);
    for(int i=1; i<=n; i++) scanf("%d",&a[i]);

    build(1,1,n);

    int q; scanf("%d",&q);
    while(q--){
        int x,y; scanf("%d%d",&x,&y);
        if(x>y)swap(x,y);
        data ans = query(1,1,n,x,y);
        ll res = max(max(ans.prefix,ans.suffix), max(ans.bestSum,ans.sum));
        printf("%lld\n",res);
    }
    return 0;
}
```

```cpp
/* SPOJ - GSS3 - Can you answer these queries III
Given a sequence A of N(N<=50000) integers between -10000 and 10000.
On this sequence you have to apply M (M <= 50000) operations:
Modify the i-th element in the sequence or
For given x y print max{Ai + Ai+1 + .. + Aj | x<=i<=j<=y }.
*/
#include<bits/stdc++.h>
using namespace std;
#define ll long long
#define MAXN 50005
#define lf nd<<1
#define rg (nd<<1)+1
#define m (int)((b+e)>>1)
#define N tree[nd]
#define L tree[lf]
#define R tree[rg]

int a[MAXN];
struct data{
    ll prefix,suffix,bestSum,sum;
}tree[4*MAXN];

data maxData(data x, data y){
    data z;
    z.prefix = max(x.prefix, x.sum + y.prefix);
    z.suffix = max(y.suffix, y.sum + x.suffix);
    z.bestSum = max(max(x.bestSum, y.bestSum), x.suffix + y.prefix);
    z.sum     = x.sum + y.sum;
    return z;
}
void build(int nd,int b,int e){
    if(b==e){ N.prefix = N.suffix = N.bestSum = N.sum = a[b]; return; }
    build(lf,b,m);
    build(rg,m+1,e);
    N = maxData(L,R);
}
void update(int nd,int b,int e,int x,int v){
    if(b==e && b==x){ N.prefix = N.suffix = N.bestSum = N.sum = a[b] = v; return; }
    if(x<=m)update(lf,b,m,x,v);
    else update(rg,m+1,e,x,v);
    N = maxData(L,R);
}
data query(int nd,int b,int e,int x,int y){
    if(b>y || e<x){ data INF; INF.prefix=INF.suffix=INF.bestSum=INF.sum=-1000000000; return INF; }
    if(b>=x && e<=y) return tree[nd];
    return maxData(query(lf,b,m,x,y), query(rg,m+1,e,x,y));
}
int main(){
    int n; scanf("%d",&n);
    for(int i=1; i<=n; i++) scanf("%d",&a[i]);

    build(1,1,n);

    int q; scanf("%d",&q);
    while(q--){
        int t,x,y; scanf("%d%d%d",&t,&x,&y);
        if(t==0){
            update(1,1,n,x,y);
        }else{
            data ans = query(1,1,n,x,y);
            ll res = max(max(ans.prefix,ans.suffix), max(ans.bestSum,ans.sum));
            printf("%lld\n",res);
        }
    }
}
```