```cpp
#include<bits/stdc++.h>
using namespace std;
#define ll long long
#define INF 10000000000000000LL
#define MAXN 10005
#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 n,a[MAXN];
struct data{
    ll prefix,suffix,infixx,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.infixx = max(max(x.infixx, y.infixx), 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 = a[b] = N.infixx = a[b] = 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 r;
        r.prefix = r.suffix = r.infixx = r.sum = -INF;
        return r;
    }

    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);
}
ll solve(int x1,int y1,int x2,int y2){
    ll ans = 0;
    if(x1==x2&&y1==y2){
        data A = query(1, 1, n, x1, y1);
        ans = max(A.prefix, max(A.suffix, A.infixx));
    }
    else if(x1==x2&&y1<y2){
        data A = query(1, 1, n, x1, y2);
        ans = max(A.prefix, max(A.suffix, A.infixx));
        data C = query(1, 1, n, y1+1, y2);
        ans = max(ans, A.suffix + C.prefix);
```

```
    }
    else if(x1<x2 && y1==y2){
        data B = query(1, 1, n, x2, y2);
        ans = max(B.prefix, max(B.suffix, B.infixx));
        data C = query(1, 1, n, x1, x2-1);
        ans = max(ans, C.suffix + B.prefix);
    }
    else if(y1==x2 && y1<y2){
        data B = query(1, 1, n, x2, y2);
        data C = query(1, 1, n, x1, y1-1);
        ans = max(B.prefix, C.suffix + B.prefix);
    }
    else if(x1<x2 && x2<y1 && y1<y2){
        data A = query(1, 1, n, x1, y1);
        data B = query(1, 1, n, x2, y2);
        data C = query(1, 1, n, x1, x2-1);
        data D = query(1, 1, n, x2, y1);
        data E = query(1, 1, n, y1+1, y2);
        ans = max(D.prefix, max(D.suffix, D.infixx));
        ans = max(ans, C.suffix + B.prefix);
        ans = max(ans, A.suffix + E.prefix);
        ans = max(ans, max(A.suffix, B.prefix));
    }
    else if(y1==x2-1){
        data A = query(1, 1, n, x1, y1);
        data B = query(1, 1, n, x2, y2);
        ans = A.suffix + B.prefix;
    }
    else if(x2-y1>1){
        data A = query(1, 1, n, x1, y1);
        data B = query(1, 1, n, x2, y2);
        data C = query(1, 1, n, y1+1, x2-1);
        ans = A.suffix + C.sum + B.prefix;
    }
    else{
        assert(0);
    }
    return ans;
}
int main()
{
    int tt; scanf("%d",&tt);
    while(tt--){
        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 x1,y1,x2,y2; scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
            ll ans = solve(x1,y1,x2,y2);
            printf("%lld\n",ans);
        }
    }

    return 0;
}
```