

```

1.  #include<bits/stdc++.h>
2.  using namespace std;
3.  #define ll long long
4.  #define INF 100000000000000LL
5.  #define MAXN 10005
6.  #define lf nd<<1
7.  #define rg (nd<<1)+1
8.  #define m (int)((b+e)>>1)
9.  #define N tree[nd]
10. #define L tree[lf]
11. #define R tree[rg]
12.
13. int n,a[MAXN];
14. struct data{
15.     ll prefix,suffix,infixx,sum;
16. }tree[4*MAXN];
17.
18. data maxData(data x, data y){
19.     data z;
20.     z.prefix = max(x.prefix, x.sum + y.prefix);
21.     z.suffix = max(y.suffix, y.sum + x.suffix);
22.     z.infixx = max(max(x.infixx, y.infixx), x.suffix + y.prefix);
23.     z.sum = x.sum + y.sum;
24.     return z;
25. }
26. void build(int nd,int b,int e){
27.     if(b==e){
28.         N.prefix = N.suffix = a[b] = N.infixx = a[b] = N.sum = a[b];
29.         return;
30.     }
31.     build(lf,b,m);
32.     build(rg,m+1,e);
33.     N = maxData(L,R);
34. }
35. data query(int nd,int b,int e,int x,int y){
36.
37.     if(b>y || e<x){
38.         data r;
39.         r.prefix = r.suffix = r.infixx = r.sum = -INF;
40.         return r;
41.     }
42.
43.     if(b>=x && e<=y){
44.         return tree[nd];
45.     }
46.
47.     data f1 = query(lf,b,m,x,y);
48.     data f2 = query(rg,m+1,e,x,y);
49.     return maxData(f1,f2);
50. }
51. ll solve(int x1,int y1,int x2,int y2){
52.     ll ans = 0;
53.     if(x1==x2&&y1==y2){
54.         data A = query(1, 1, n, x1, y1);
55.         ans = max(A.prefix, max(A.suffix, A.infixx));
56.     }
57.     else if(x1==x2&&y1<y2){
58.         data A = query(1, 1, n, x1, y2);
59.         ans = max(A.prefix, max(A.suffix, A.infixx));
60.         data C = query(1, 1, n, y1+1, y2);
61.         ans = max(ans, A.suffix + C.prefix);

```

```

62.     }
63.     else if(x1<x2 && y1==y2){
64.         data B = query(1, 1, n, x2, y2);
65.         ans = max(B.prefix, max(B.suffix, B.infixx));
66.         data C = query(1, 1, n, x1, x2-1);
67.         ans = max(ans, C.suffix + B.prefix);
68.     }
69.     else if(y1==x2 && y1<y2){
70.         data B = query(1, 1, n, x2, y2);
71.         data C = query(1, 1, n, x1, y1-1);
72.         ans = max(B.prefix, C.suffix + B.prefix);
73.     }
74.     else if(x1<x2 && x2<y1 && y1<y2){
75.         data A = query(1, 1, n, x1, y1);
76.         data B = query(1, 1, n, x2, y2);
77.         data C = query(1, 1, n, x1, x2-1);
78.         data D = query(1, 1, n, x2, y1);
79.         data E = query(1, 1, n, y1+1, y2);
80.         ans = max(D.prefix, max(D.suffix, D.infixx));
81.         ans = max(ans, C.suffix + B.prefix);
82.         ans = max(ans, A.suffix + E.prefix);
83.         ans = max(ans, max(A.suffix, B.prefix));
84.     }
85.     else if(y1==x2-1){
86.         data A = query(1, 1, n, x1, y1);
87.         data B = query(1, 1, n, x2, y2);
88.         ans = A.suffix + B.prefix;
89.     }
90.     else if(x2-y1>1){
91.         data A = query(1, 1, n, x1, y1);
92.         data B = query(1, 1, n, x2, y2);
93.         data C = query(1, 1, n, y1+1, x2-1);
94.         ans = A.suffix + C.sum + B.prefix;
95.     }
96.     else{
97.         assert(0);
98.     }
99.     return ans;
100. }
101. int main()
102. {
103.     int tt; scanf("%d",&tt);
104.     while(tt--){
105.         scanf("%d",&n);
106.         for(int i=1; i<=n; i++){
107.             scanf("%d",&a[i]);
108.         }
109.
110.         build(1,1,n);
111.
112.         int q; scanf("%d",&q);
113.         while(q--){
114.             int x1,y1,x2,y2; scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
115.             ll ans = solve(x1,y1,x2,y2);
116.             printf("%lld\n",ans);
117.         }
118.     }
119.
120.     return 0;
121. }

```