

```

1.  #include<bits/stdc++.h>
2.  using namespace std;
3.  #define ll long long
4.  #define INF 1000000000000000LL
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.     if(b==x && e==y){
37.         return tree[nd];
38.     }
39.     data f;
40.     if(y<=m){
41.         f = query(lf,b,m,x,y);
42.     }
43.     else if(x>m){
44.         f = query(rg,m+1,e,x,y);
45.     }
46.     else{
47.         data f1 = query(lf,b,m,x,m);
48.         data f2 = query(rg,m+1,e,m+1,y);
49.         f = maxData(f1,f2);
50.     }
51.     return f;
52. }
53. ll solve(int x1,int y1,int x2,int y2){
54.     ll ans = -INF;
55.     if(x1==x2&&y1==y2){
56.         data A = query(1, 1, n, x1, y2);
57.         ans = max(A.prefix, max(A.suffix, A.infixx));
58.     }
59.     else if(x1==x2&&y1<y2){
60.         data A = query(1, 1, n, x1, y2);
61.         ans = max(A.prefix, max(A.suffix, A.infixx));
62.         data C = query(1, 1, n, y1+1, y2);

```

```

63.     ans = max(ans, A.suffix + C.prefix);
64. }
65. else if(x1<x2 && y1==y2){
66.     data B = query(1, 1, n, x2, y2);
67.     ans = max(B.prefix, max(B.suffix, B.infixx));
68.     data C = query(1, 1, n, x1, x2-1);
69.     ans = max(ans, C.suffix + B.prefix);
70. }
71. else if(y1==x2 && y1<y2){
72.     data B = query(1, 1, n, x2, y2);
73.     data C = query(1, 1, n, x1, y1-1);
74.     ans = max(B.prefix, C.suffix + B.prefix);
75. }
76. else if(x1<x2 && x2<y1 && y1<y2){
77.     data A = query(1, 1, n, x1, y1);
78.     data B = query(1, 1, n, x2, y2);
79.     data C = query(1, 1, n, x1, x2-1);
80.     data D = query(1, 1, n, x2, y1);
81.     data E = query(1, 1, n, y1+1, y2);
82.     ans = max(D.prefix, max(D.suffix, D.infixx));
83.     ans = max(ans, C.suffix + B.prefix);
84.     ans = max(ans, A.suffix + E.prefix);
85.     ans = max(ans, max(A.suffix, B.prefix));
86. }
87. else if(y1==x2-1){
88.     data A = query(1, 1, n, x1, y1);
89.     data B = query(1, 1, n, x2, y2);
90.     ans = A.suffix + B.prefix;
91. }
92. else if(x2-y1>1){
93.     data A = query(1, 1, n, x1, y1);
94.     data B = query(1, 1, n, x2, y2);
95.     data C = query(1, 1, n, y1+1, x2-1);
96.     ans = A.suffix + C.sum + B.prefix;
97. }
98. else{
99.     assert(0);
100. }
101.
102. if(ans>=-)
103.     return ans;
104. }
105. int main()
106. {
107.     int tt; scanf("%d",&tt);
108.     while(tt--){
109.         scanf("%d",&n);
110.         for(int i=1; i<=n; i++){
111.             scanf("%d",&a[i]);
112.         }
113.
114.         build(1,1,n);
115.
116.         int q; scanf("%d",&q);
117.         while(q--){
118.             int x1,y1,x2,y2; scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
119.             ll ans = solve(x1,y1,x2,y2);
120.             printf("%lld\n",ans);
121.         }
122.     }
123.
124.     return 0;
125. }

```

