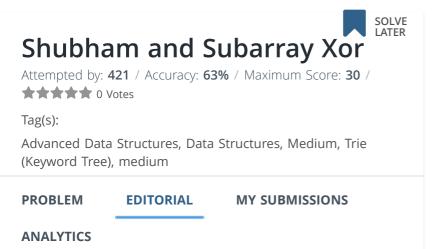
CHALLENGES

PRACTICE

COMPANIES

All Tracks > Data Structures > Advanced Data Structures > Trie (Keyword Tree) > Problem



One of the standard applications of the trie data structure is that finding maximum xor value of 2 integers given some integer values. This would be quite helpful in solving this particular question.

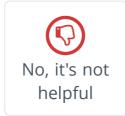
We iterate the array in reverse order. Suppose we are at index i. We consider it to be the right end point of one of the subarrays. We then loop over j where $1 \leq j \leq i$ which is the left end point of the subarray. We consider we have inserted sum of all subarrays [l,r] where $i \leq l \leq r \leq n$ in a trie. So we can compute the maximum xor value of ($\operatorname{sum}[i,j] \operatorname{xor} x$) where x is a value in trie. After iterating over all j, we insert in the trie sum of subarrays starting at index i.

Please have a look at setter/tester code for implementation details.

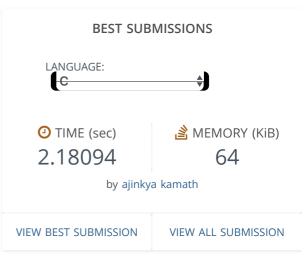
Complexity: $O(n^2 * log(sum(a_i)))$

IS THIS EDITORIAL HELPFUL?





7 developer(s) found this editorial helpful.

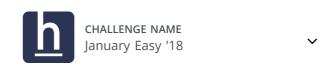


CONTRIBUTOR





THIS PROBLEM WAS ASKED IN



SOCIAL SHARE



Author Solution by shubham goyal

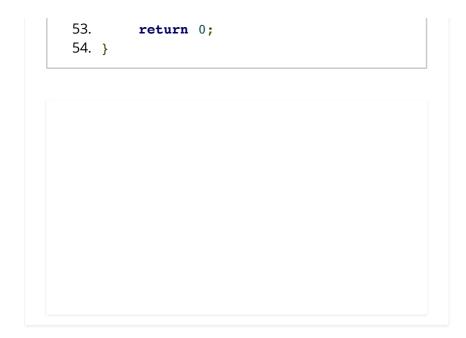
```
1. //let's do it
 2. #include <iostream>
 3. #include <bits/stdc++.h>
 4. using namespace std;
 5. #define 11 long long int
 6. #define inf 100000000000
 7. #define mod 1000000007
 8. #define pb push back
 9. #define mp make pair
10. #define all(v) v.begin(), v.end()
11. #define S second
12. #define F first
13. #define boost1
   ios::sync with stdio(false);
14. #define boost2 cin.tie(0);
15. #define mem(a,val)
   memset(a,val,sizeof a)
16. #define endl "\n"
17. #define maxn 820000
18.
19. 11
   tr[8*maxn][2],nn=1,cnt[8*maxn][2],arr
   [905],sum[905];
20. void add(ll a) {
21.
       11 t = 1;
22.
        for (11 i = 31; i >= 0; --i) {
23.
             int now = (a >> i) & 1;
24.
             if (!tr[t][now]) {
25.
                 tr[t][now] = ++nn;
26.
27.
            cnt[t][now]++;
28.
            t = tr[t][now];
29.
        }
30. }
31. ll getMax(ll a) {
32.
        11 t = 1, res = 0;
        for (11 i = 31; i >= 0; --i) {
33.
34.
             11 \text{ now} = (a >> i) \& 1;
35.
             now = !now;
36.
             if (tr[t][now] &&
   cnt[t][now]) {
37.
                 t = tr[t][now];
38.
                 res += (1 << i) * 1;
39.
             } else {
40.
                 t = tr[t][!now];
41.
             }
```

```
42.
43.
         return res;
44. }
45. inline ll getsum(ll l, ll r)
46. {
47.
             assert(1<=r);</pre>
48.
             return sum[r]-sum[l-1];
49. }
50. int main()
51. {
52.
             boost1; boost2;
53.
             11 i,j,n,x,y,val,ans;
54.
             cin>>n;
55.
             for(i=1;i<=n;i++)</pre>
56.
             cin>>arr[i];
57.
             sum[1]=arr[1];
58.
             for(i=2;i<=n;i++)</pre>
59.
             sum[i]=sum[i-1]+arr[i];
60.
             add(arr[n]);
61.
             ans=0;
62.
             for(i=n-1;i>=1;i--)
63.
64.
                      for(j=i; j>=1; j--)
65.
66.
    val=getsum(j,i);
67.
    ans=max(ans,getMax(val));
68.
69.
                      for(j=i;j<=n;j++)</pre>
70.
71.
    val=getsum(i,j);
72.
                                add(val);
73.
                       }
74.
75.
             cout<<ans;
76.
77.
             return 0;
78. }
```

Tester Solution by Mitesh Agrawal

```
    //Mitesh Agrawal
    #include <bits/stdc++.h>
    using namespace std;
    4.
    #define next _nxt
    6.
```

```
7. const int N = 10000005;
 8. int sz = 0, next[2][N], arr[905],
   sum[905];
 9. bool created[N];
10.
11. void insert (int s) {
12.
        int v = 0;
13.
        for (int i = 30; i >= 0; i--) {
14.
             int c = (s >> i) & 1;
15.
             if (!created[next[c][v]]) {
16.
                  next[c][v] = ++sz;
17.
                  created[sz] = true;
18.
19.
             v = next[c][v];
20.
        }
21. }
22.
23. int search (int tmp) {
24.
        int v = 0, ans = 0;
25.
        for (int i = 30; i >= 0; i--) {
26.
             int c = (tmp >> i) & 1;
27.
             if(created[next[1 ^ c][v]]){
28.
                  ans |= ((1 ^c) << i);
29.
                  v = next[1 ^ c][v];
30.
31.
             else{
32.
                  ans |= (c << i);
33.
                  v = next[c][v];
34.
             }
35.
        }
36.
        return ans;
37. }
38.
39. int main(){
40.
        int i,j,n,maxi = 0,curr;
41.
        scanf("%d", &n);
42.
        for(i = 1; i <= n; i++){
43.
             scanf("%d", &arr[i]);
44.
             sum[i] = sum[i - 1] +
   arr[i];
45.
46.
        for(i = 1; i <= n; i++){
47.
             for(j = 1; j <= i; j++)</pre>
48.
                  insert(sum[i] - sum[j -
   1]);
49.
             for(j = i + 1; j <= n; j++)</pre>
50.
                 maxi = max(maxi, (sum[j]
   - sum[i]) ^ search(sum[j] - sum[i]));
51.
52.
        printf("%d\n", maxi);
```



About Us Innovation Management Talent Assessment University Program Developers Wiki Blog

Press Careers Reach Us

h

Site Language: English 🛊 | Terms and Conditions | Privacy |© 2018 HackerEarth