

CHALLENGES

PRACTICE

COMPANIES

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

Shubham and Subarray Xor

Attempted by: 421 / Accuracy: 63% / Maximum Score: 30 /

★★★★★ 0 Votes

Tag(s):

Advanced Data Structures, Data Structures, Medium, Trie (Keyword Tree), medium

PROBLEM

EDITORIAL

MY SUBMISSIONS

ANALYTICS

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 $(\text{sum}[i, j] \text{ xor } x)$ 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: $O(n^2 * \log(\text{sum}(a_i)))$

IS THIS EDITORIAL HELPFUL?



Yes, it's helpful



No, it's not helpful

7 developer(s) found this editorial helpful.

SOLVE
LATER

BEST SUBMISSIONS

LANGUAGE:



TIME (sec)

2.18094

MEMORY (KiB)

64

by ajinkya kamath

VIEW BEST SUBMISSION

VIEW ALL SUBMISSION

CONTRIBUTOR



AUTHOR

shubham goyal



TESTER

Mitesh Agrawal

THIS PROBLEM WAS ASKED IN



CHALLENGE NAME

January Easy '18



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 ll long long int
6. #define inf 1000000000000
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. ll
    tr[8*maxn][2],nn=1,cnt[8*maxn][2],arr
    [905],sum[905];
20. void add(ll a) {
21.     ll t = 1;
22.     for (ll 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.     ll t = 1, res = 0;
33.     for (ll i = 31; i >= 0; --i) {
34.         ll 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(l<=r);
48.     return sum[r]-sum[l-1];
49. }
50. int main()
51. {
52.     boost1;boost2;
53.     ll i,j,n,x,y,val,ans;
54.     cin>>n;
55.     for(i=1;i<=n;i++)
56.         cin>>arr[i];
57.     sum[1]=arr[1];
58.     for(i=2;i<=n;i++)
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.
67.             val=getsum(j,i);
68.             ans=max(ans, getMax(val));
69.         }
70.         for(j=i;j<=n;j++)
71.         {
72.             val=getsum(i,j);
73.             add(val);
74.         }
75.     }
76.     cout<<ans;
77.     return 0;
78. }

```

Tester Solution by [Mitesh Agrawal](#)

```

1. //Mitesh Agrawal
2. #include <bits/stdc++.h>
3. using namespace std;
4.
5. #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++)
48.             insert(sum[i] - sum[j -
1]);
49.         for(j = i + 1; j <= n; j++)
50.             maxi = max(maxi, (sum[j]
- sum[i]) ^ search(sum[j] - sum[i]));
51.     }
52.     printf("%d\n", maxi);
```

```
53.     return 0;  
54. }
```

[About Us](#)[Innovation Management](#)[Talent Assessment](#)[University Program](#)[Developers Wiki](#)[Blog](#)[Press](#)[Careers](#)[Reach Us](#)

Site Language: English | [Terms and Conditions](#) | [Privacy](#) | © 2018 HackerEarth