Problem Solving 6

Tuesday, 31 August 2021 8:54 PM

① Bulbs Uber

N light bulbs
Each bulb has a switch

Due to faulty wiring, a switch also changes the state of all the bulbs to right of it

Ans: min no. of switches to be pressed to get them all ON

0 1 0 1 0 0 1 → OFF ON

1 1 1 1 1 1 1

0 1 0 1 ← ans[]

1 0 1 0

First / Last

Move 1

1 1 0 1
1 1 1 0
1 1 1 1

1 1 0 0

0 0 1 0
0 0 1 1
0 0 0 0
1 1 1 1

1 1 0 1
1 1 0 0
1 1 1 1

i [i+1 → n]

i th

1 0 0 1 0 1 1

0 0 0 0
1 1 1 1

Left to right

→ TC → O(N²) N + N-1 + (N-2) . . .

→ if bulb is on → continue

→ if bulb is off → toggle that switch and toggle all switches to the right of it

i = 0 count = 3 count % 4

0 1 1 1 0 1

ans[i] = 1

1 0 0 0 1 0

ans[i] == 1 but count % 2 == 0

Val == 1

original: 0 1 1 1 0 1

1 → 0 → 1 → 0

1 → 0 → 1 → 0 → 1

0 1 1 1 0 1 count = 0 → 1 → 2 → 3 → 4

TC O(N) SC O(1)

0 1 1 1 0 1
1 0 0 0 1 0
1 1 1 1 0 1
1 1 1 1 1 0
1 1 1 1 1 1

0 1 1 1 0 1 count = 0

Val = 0 count = 0 → 0
Val = 1 count = 1 → 0
Val = 1 count = 2 → 1
Val = 0 count = 2 → 0
Val = 1 count = 3 → 0
count = 4

0 0 0 0

Val = 0 count = 0

Val = 0 count = 1 → 1

1 1 1 1
0 0 0 0
0 0 0

0 0 1 1 1 0

count = 0

1 1 1 1 1

count = 0
for i = 0 → n-1
  if (ans[i] == 0 && count % 2 == 0)
    count++
  else if (ans[i] == 1 and count % 2 == 1)
    count++

0 1 1 0 1
1 0 0 1 0
1 1 1 0 1
1 1 1 1 0
1 1 1 1 1

Q An array of N size is given to you
ans[i] → 0/1

Ques
↳ 2 types of queries
- 1st type of query → XOR [l . . r]
↳ 2nd type → count of 0's from [l . . . r]

O(QN)

1 2 3 4 5 6 7 8
0 0 1 1 0 1 1 0

7 - 2 + 1

(-1)
1 → 2 7 → 0 ^ 1 ^ 1 ^ 0 ^ 1 ^ 1 = 0
2 → 1 6 → 3

0 ^ 1

0 1 2 3 4 5 6 7 8
0 0 0 1 0 0 1 0 0

pref XOR[i] = ans[1] ^ pref XOR[i-1]

pref XOR[i] = XOR from 1 index to ith

ans[1] ^ ans[2] ^ ans[3] . . . ans[i]

0 ^ 0 → 0

2 → 7

0 ^ 1 ^ 1 ^ 0 ^ 1 ^ 0

1 2 3

x ^ x = 0

ans[1]
= ans[2] ^ ans[3] . . . ans[i]

SOLNS

① brute force
→ loop from L to R TC O(Q*N) SC → O(1)

② 1st Query → Prefix XOR Array
2nd Query → count of zero

ans[1] ^ ans[2] . . ans[i]

pref XOR → XOR of all values from 1st index to ith index

ans[1] ^ ans[2] . . ans[l-1]

pref XOR[r] ^ pref XOR[l-1]

XOR[l ^ . . . r] 0 - 1 = -1

r - l + 1

pref sum[r] - pref[l-1]
→ sum up till r th index, subtract no. of ones from the length of subarray being queried

1 . . . 2 . . l-1] l . . . . r

0 1 2 3
0 1 0 1 1 0

0 3

0 1 1 2 3 3

r - l + 1 ← - (ans[r] - ans[l-1])

4 - 2 = 2

③ - count of ones
prefix sum / prefix count of ones

Tiles a×a N*M 10:30

Tiles N×M grid

N = 7 a = 3 M = 4

a = 2

N/a

6/2 = 3

ceil(m/a) × ceil(n/a)

N = 7 3 × 1

a×a 2×2 2×2

MIN No. of tiles require to cover the whole grid

N items → n
x items atmost to 1 child

ceil(N/x)

Q Given array of size N containing only 0's or 1's. Find count of all subarrays having OR value of subarray as 1

[1 0 1] → 5
[1] → 1
[0] → 0
[1] → 1
[1, 0] → 1
[0, 1] → 1
[1, 0, 1] → 1

a[l : r]

6 - 1 = 5 subarray →

N*(N+1)/2 3×4/2 = 6

3 + 2 + 1 = 6

0|0 = 0
0|1 = 1
1|0 = 1
1|1 = 1

i = 0, 1 i = 1
i = 0, 2
0, 1

N + N-1

[5 4 6]
5 → 5
4 → 4
6 → 6
5 4 → 5
4 6 → 6
5 4 6 → 7

101
100
101
110

101 | 100 | 110
111 → 7

OR value of every subarray
↓
Give sum of all subarray

5 + 4 + 6 + 5 + 6 + 7 = .

→ 32 * N

N*(N+1)/2

1 [0 0 0] 0 0 0
1 [1 1] 1 1

OR value:

1 0 0 1
0 0 0

4*5/2 = 10 - 2 = 8

1 0 0 1
0 → 0
0 → 0
→ 1
1 0 → 1
0 0 → 0
0 1 → 1
1 0 0 → 1
0 0 1 → 1
1 0 0 1 → 1

N*(N+1)/2

7

N*(N+1)/2 - Σ (x*(x+1)/2)

0 0 0 0

4*5/2 = 10

1 0 0 1 0 0 1

(7*(7+1))/2 - (3+3)

1 1 0 0 1 0 1 0 0 0 0 1 0 1 1 1 1

0 0 0 0