0 1 2 3 4 ... 43 44

16 12

Q1 Given an array, Build **leftmax** array

leftmax[i] = max arr value in range [0, i]

	0	1	2	3	4	5	6	7
A:	-3	6	2	4	5	2	8	-9
leftmax	-3	6	6	6	6	6	8	8

a_0 a_1 a_2

leftmax[0] = a_0

leftmax[1] = max(a_0, a_1)

max(leftmax$_0$, a_1)

leftmax[2] = max(a_0, a_1, a_2)

max(leftmax$_1$, a_2)

leftmax[i] = max(leftmax$_{i-1}$, a_i)

90 %. 85 %.

```
int leftmax[N]
leftmax[0] = a[0]
for(int i=1; i<n; i++){
    leftmax[i] = max(leftmax[i-1], a[i])
}
```

TC: $O(N)$ SC: $O(n)$

Same Assignment

Q Leaders ⟹

How many elements $a[i] > lm[i-1]$

	0	1	2	3	4	5	6	7
A:	-3	6	2	4	5	2	8	-9
leftmax	-3	6	6	6	6	6	8	8

ans = 3

Given: Assume a[0] is a leader

```
c = 1
for( i=1; i<n; i++){
    if( a[i] > leftmax(i-1) )
        c++
}
```

- y

return c

Amazon

Q2 Given a string of lowercase char, return the count of pairs (i,j) st

indexes.

$i < j$

$s[i] = 'a'$

$s[j] = 'g'$

0 1 2 3 4 5

S: a b e g a g

0,3 0,5 4,5

Pairs ⇒

ans = 3

0 1 2 3 4 5 6

S: a c g d g a g

Pairs ⇒ 0,2 0,4 0,6 5,6

ans = 4

Observation: every 'g' will make pair with all the 'a' on the left

	0	1	2	3	4	5	6	7
Ex	a	c	b	a	g	k	a	g
count_a = 0	1	1	1	2	2	2	3	3
ans = 0	0	0	0	0	0+2 = 2	2	2	2+3 = 5

Code

```
int ans = 0, count_a = 0
for (int i=0; i<n; i++) {
    if (s[i] == 'a')
        count_a++
    else if (s[i] == 'g')
        ans = ans + count_a
}
return ans
```

TC: $O(n)$

SC: $O(1)$

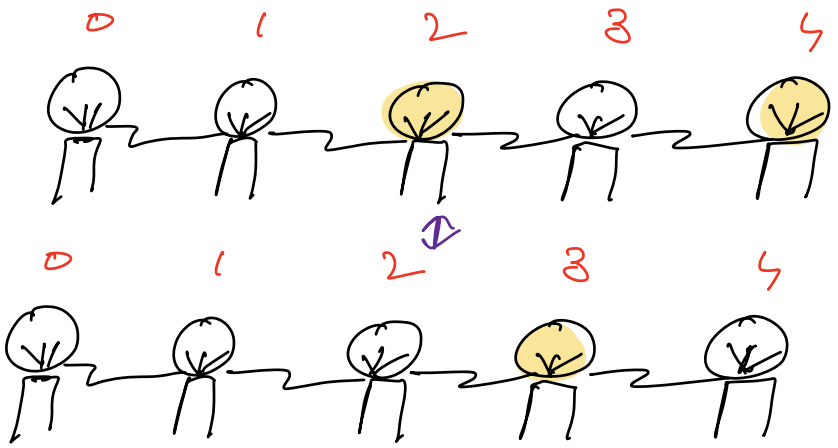
Directi Top PBC

Q3 N Bulbs ON/OFF

Given N bulbs & their state, each bulb has switch that will flip all bulbs on the right

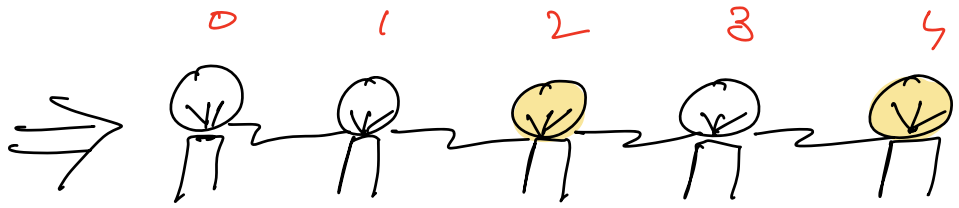
ON ⇆ OFF

Min switches required for all bulbs to be ON



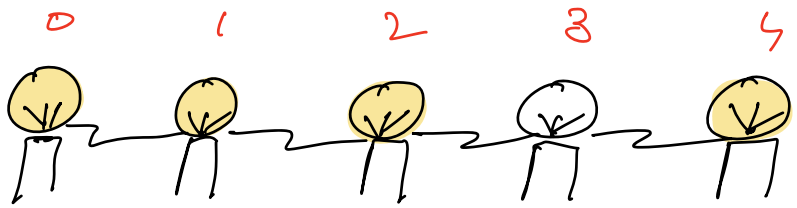
Obs: 0 is only affected by 0

Obs: Go from left to right & flip bulb if required

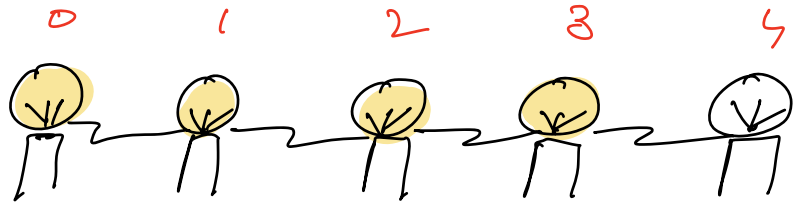


i = 0

i = 2



i = 3



i = 4

ans = 4

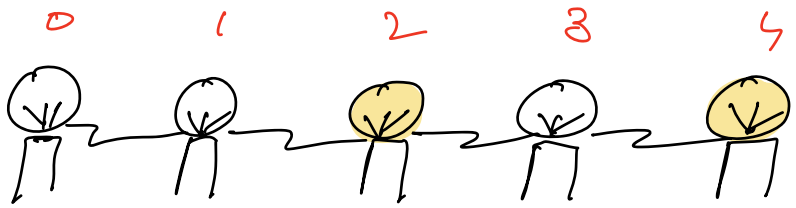
Sol^n ⇒ Iterate through the bulbs & flip if required.
On flip, everything on the right is also flipped

Obs: If a bulb is flipped 2 times ⇒ ON → OFF → ON
OFF → ON → OFF

4 times ⇒ same
5 times ⇒ diff
1331 times ⇒ diff

odd ⇒ diff
even ⇒ same

Just need to maintain how many flips have been done



c=0 1 1 2 3 4

Code

```
int switches (int arr[]){
    int c=0
    for(i=0; i<n; i++){
        // find actual state
        if( c%2 == 1 )
            a[i] = 1-a[i]   // flip
        if( a[i] == 0 )
            c++
    }
    return c
}
```

TC: O(N) SC: O(1)

[done]

0 → 1

1 → 0

1 - x

```
for(i=0; i<n; i++) {
    if (a[i] == 1)
        continue
    else                    // a[i] = 0
        flip++
        for(j=i; j<n; j++) {
            a[i] = 1-a[i]   // flip
        }
}
```

TC : $O(n^2)$

FLIP

$0 \rightarrow 1$

$1 \rightarrow 0$

1 − x	x	ans
	0	1
1		0

$N \to N/2 \quad n/4 \; - - - -$