

## Todays Content

- a) Count Pairs
- b) leaders in array
- c) N Bulbs

### Q8) Count Pairs 'ag':

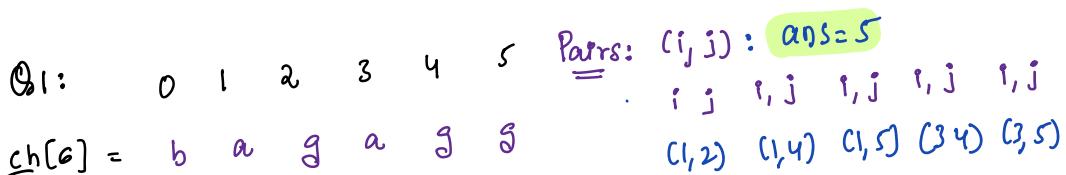
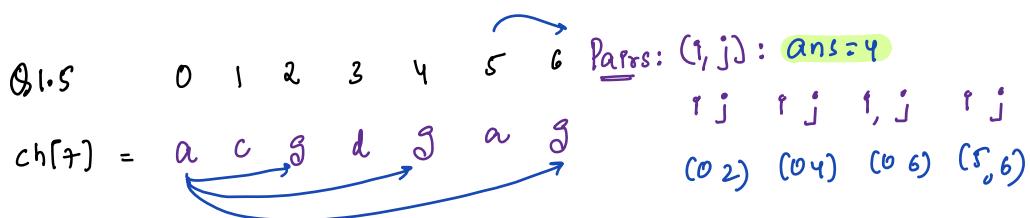
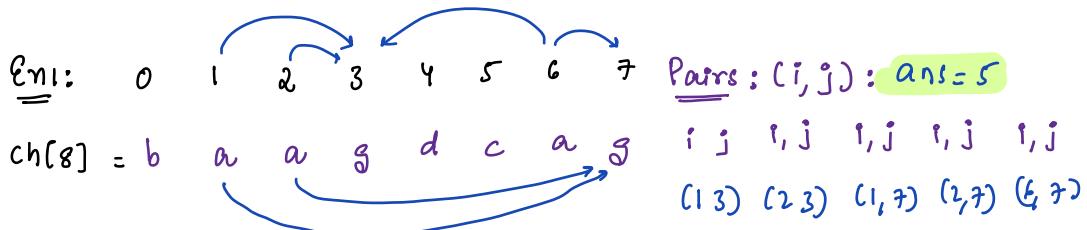
Given a char  $ch[N]$ , calculate no: of pairs indices =  $i, j$  such that

$i < j \text{ and } ch[i] == 'a' \text{ and } ch[j] == 'g'$   $\forall i, j$  characters are lower case

Constraints:

$1 \leq N \leq 10^5$  // length of  $ch[]$  at max =  $10^5$

'a'  $< ch[i] <= 'z'$



Idea: Generate all pairs such that  $i < j$ , & count if  $s[i] == 'a'$  &  $s[j] == 'g'$

N=6:  $i=0: (\cancel{\emptyset}, 1) \ (\cancel{0}, 2) \ (\cancel{0}, 3) \ (\cancel{0}, 4) \ (\cancel{0}, 5) * \text{Skip}$

$i=1: (1, 2) \ (1, 3) \ (1, 4) \ (1, 5)$

$i=2: (\cancel{2}, 3) \ (2, 4) \ (\cancel{2}, 5) * \text{Skip}$

$i=3: (3, 4) \ (3, 5)$

$i=4: (\cancel{4}, 5)$

Code: 9:25 PM

```
int pairs( char ch[] ) { TC: O(N^2) SC: 16 Bytes
    int N = ch.length, c = 0;
    for( int i=0; i < N; i++ ) {
        for( int j = i+1; j < N; j++ ) {
            // (i,j) : check as
            if( ch[i] == 'a' && ch[j] == 'g' ) {
                c = c + 1
            }
        }
    }
    return c;
}
```

Obs1:  $N = 10^5, O(N^2) = [10^5]^2 = 10^{10} \text{ TLE}$

Obs2:  $s[i] == 'a'$

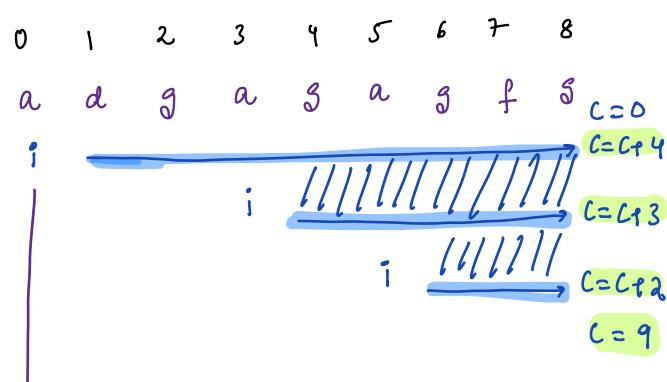
```
int pairs( char ch[] ) {
    int N = ch.length, c = 0;
    for( int i=0; i < N; i++ ) {
        if( ch[i] == 'a' ) {
            for( int j = i+1; j < N; j++ ) {
                // (i,j) : check as
                if( ch[j] == 'g' ) {
                    c = c + 1
                }
            }
        }
    }
    return c;
}
```

TC:  $O(N^2)$  SC:  $O(1)$

TC Table: TODO & End

i	j : [ ]	Iterations

for( int i=0; i < N; i++ ) { i = N  $\Rightarrow$  Stop  
*i = N-1, j = N; j < N; j++* } \*  
 It won't enter loop.

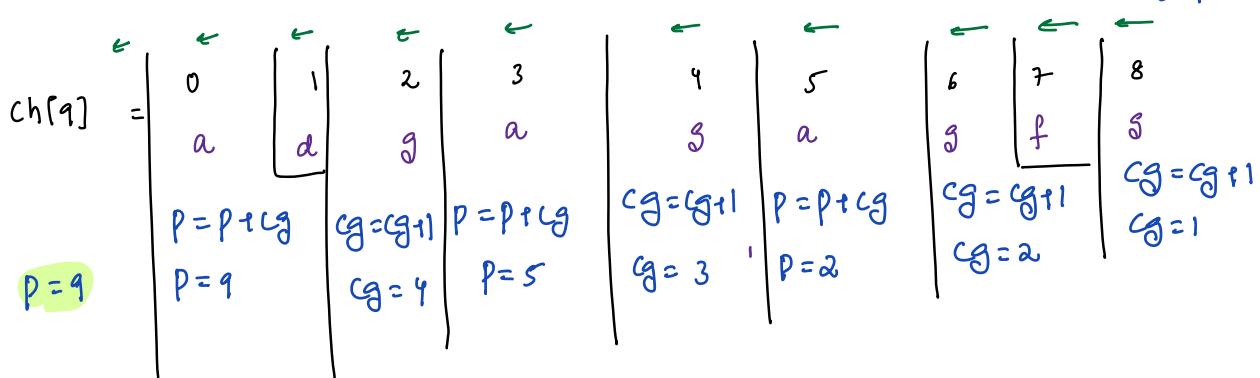


Obs: if  $ch[i] == 'a'$ :

iterating on right & calculating  
no: of 'g's

## Optimization Idea 9:35

obs1: Iterate from right to left & keep count of no: of g's



int Pairs( char ch[] ) { TC: O(N) SC: O(1) }

$$\text{int } c_0 = 0, p = 0 \quad \rightarrow \quad N \approx 10^5 : 10^3 \times 10^0$$

```
int N = ch.length;
```

```
for(int i = N-1; i >= 0; i -> {
```

If ( ch[i] == 'g') { cg++ }

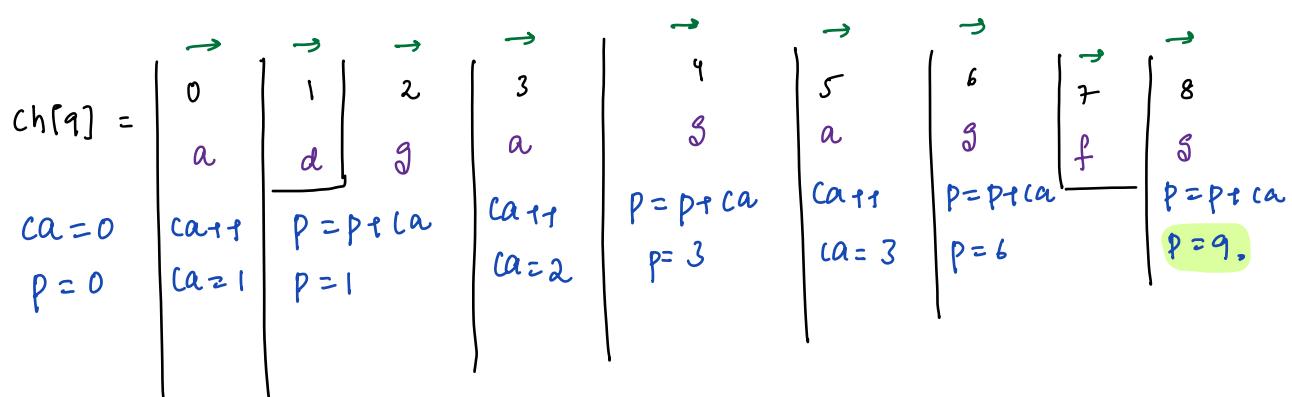
else if (ch[i] == 'a') {

$$P = P + Cg$$

return p;

obs2: For every  $-g$  get count of 'a's on left

a a a a,  $\bar{g} : \text{TODO}$



## Q8) Leaders in a Array

Given an  $ar[N]$ , you have to find no. of leaders in  $ar[?]$

Note1:  $ar[i]$  is said leader,

: if it's greater than  $\text{man}$  of all elements on left from  $[0, i-1]$

Note2:  $ar[0]$  is considered as leader

Constraints:

$$1 \leq N \leq 10^5 \quad 1 \leq ar[i] \leq 10^9$$

Ex1:    0    1    2    3    4    5    6    7    ans = 5

$ar[8]:$     3    2    4    5    2    7    -1    15

$\text{manleft}:$  - 3    3    4    5    5    7    7

Q2:    0    1    2    3    4    5    ans = 3

$ar[6]:$     4    2    3    9    7    10

$\text{manleft}:$  - 4    4    4    9    9

Idea: For every  $ar[i]$ , iterate on left & get man, check if it's leader or not.

`int leaders(int ar[])` TC: O(N<sup>2</sup>) SC: O(1)

`int N = ar.length, l=1/ar[0]`

`for(int i=1; i<N; i++) {`

`// ar[i] leader > man{0..i-1}`

`int m = Integer.MIN_VALUE`

`for(int j=0; j < i; j++) {`

`if(m < ar[j]) { m = ar[j]; }`

`// m is man from [0..i-1]`

`if(ar[i] > m) { l = l+1; }`

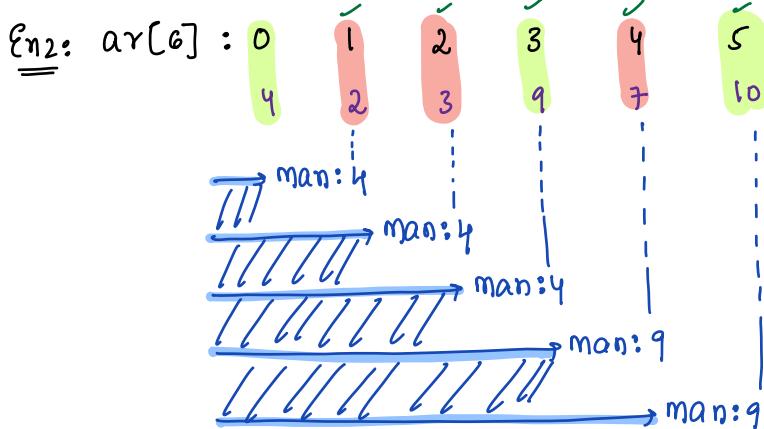
`}`

$$N = 10^5 = 10^8 \text{ TLE}$$

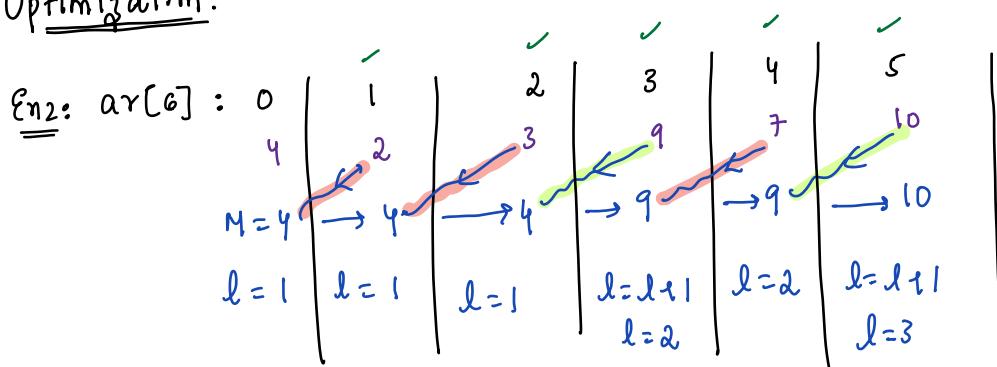
TC Table: TODO

i	j	Iterations

Tracing: 10hr 15mins



Optimization:



int leaders(int ar[]){ TC: O(N) SC: O(1)

```

int N = ar.length;
int M = ar[0], l = 1;
for(int i = 1; i < N; i++) {
    // Check if ar[i] leader?
    if(ar[i] > M) {
        l = l + 1
        M = ar[i]
    }
}
return l;
}

```

N Bulbs: 10hr : 35min

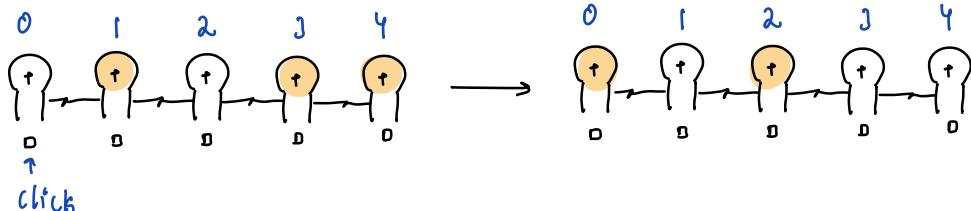
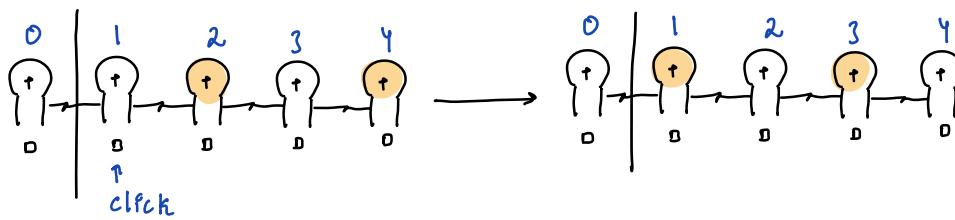
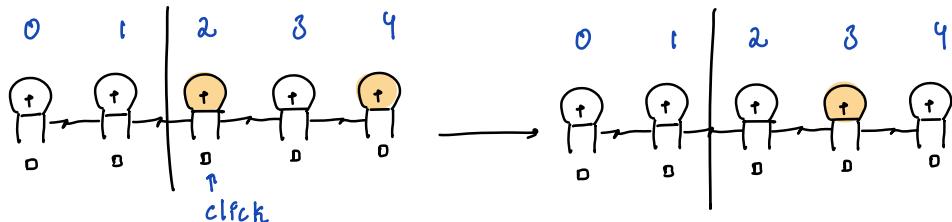
ON/OFF

Given N bulbs & their initial state, each bulb has a switch associated to it.

If we click on a switch

: Every bulb on right, including current bulb is flipped : On  $\leftrightarrow$  OFF

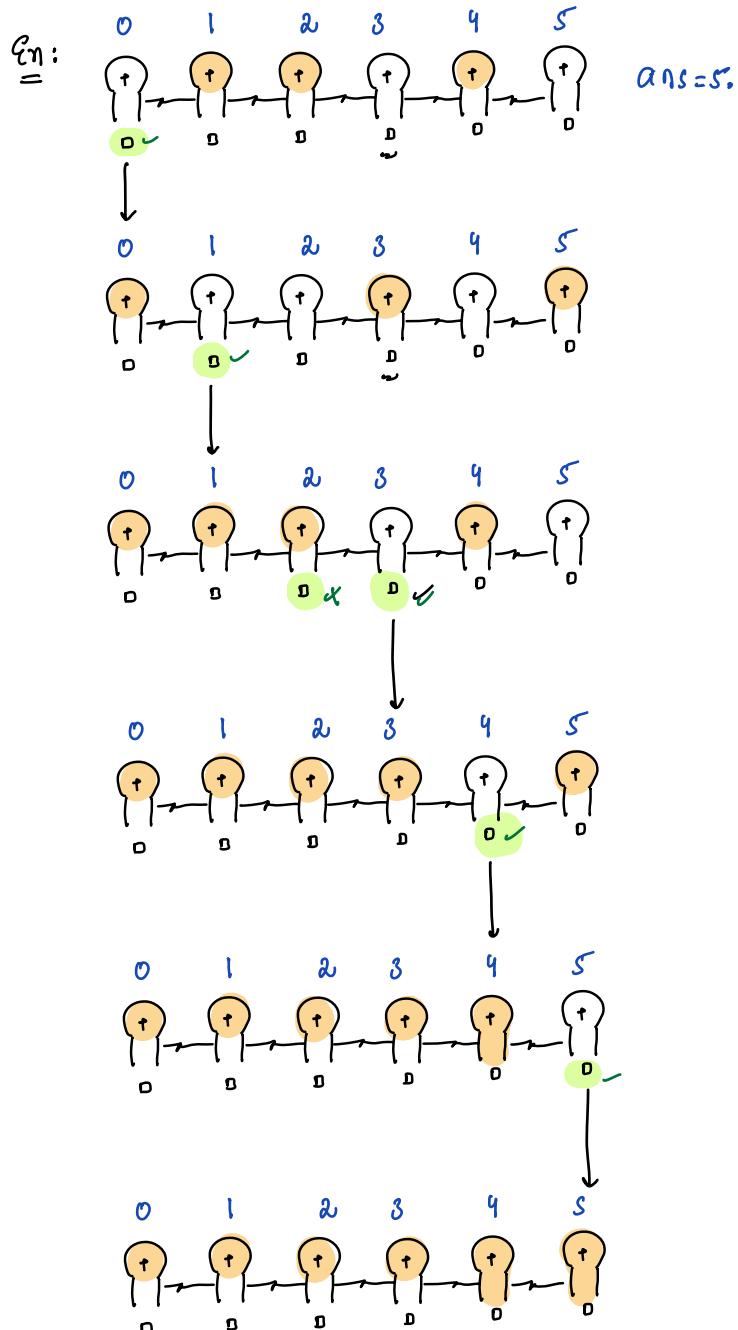
Eg:



Constraints:  $1 \leq N \leq 10^5$

Q : Min no: of time we need click on switch so that all bulbs ON

Idea: Iterate from left to right, if a bulb is off switch on



// bulb[i] = 0 : i<sup>th</sup> bulb is off

// bulb[i] = 1 : i<sup>th</sup> bulb is on

### PseudoCode:

```
int minSwitches (int bulb[]) { T: O(N2) SC: O(1)
```

    int c=0, N=bulb.length;       $\hookrightarrow N=10^5 = [10^5]^2 = 10^{10}$  TLE.

```
    for (int i=0; i < N; i++) {
```

        // When to switch bulb[i]

        if (bulb[i]==0) { [All right bulbs will get flipped]

            // Switch it & [right of index i = [i+1, i+2... N-1]]

            bulb[i]=1

            c=c+1;

```
            for (int j=i+1; j < N; j++) {
```

                if (bulb[j]==1) { bulb[j]=0 }

                else { bulb[j]=1 }

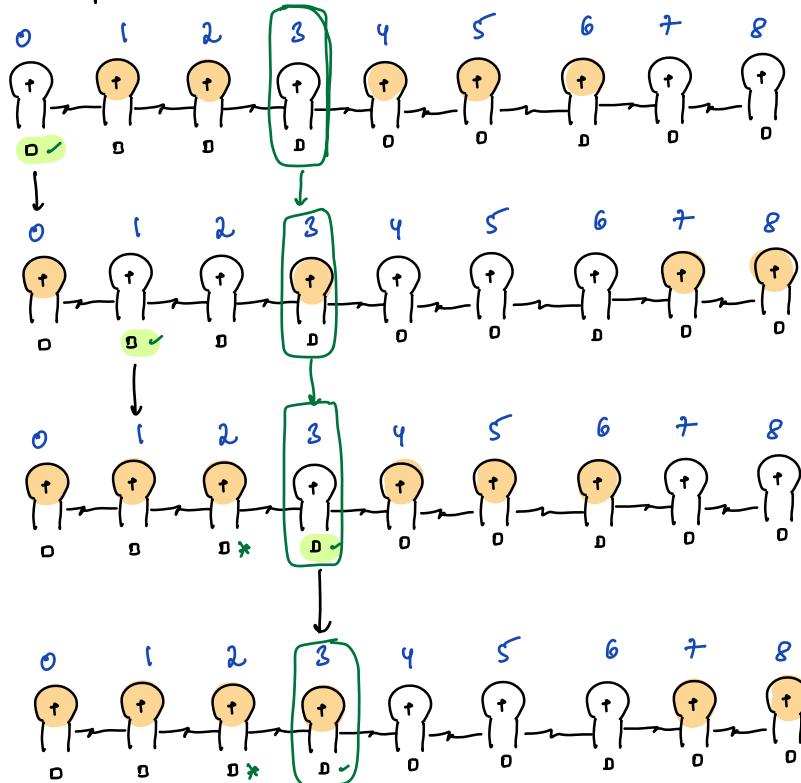
}

}

.

return c;

Idea: 10:58pm



obs:

Initial bulb 3: OFF

Flip it 3 times

Final State

$\text{OFF} \rightarrow \text{ON} \rightarrow \text{OFF} \rightarrow \text{ON}$

Quizzes: Q3:  $\text{ON} \xrightarrow{1} \text{OFF} \xrightarrow{2} \text{ON} \xrightarrow{3} \text{OFF}$ : Final OFF

Q4:  $\text{ON} \xrightarrow{1} \text{OFF} \xrightarrow{2} \text{ON} \xrightarrow{3} \text{OFF} \xrightarrow{4} \text{ON}$ : Final ON

Q5:  $\text{OFF} \xrightarrow{1} \text{ON} \xrightarrow{2} \text{OFF} \xrightarrow{3} \text{ON} \xrightarrow{4} \text{OFF}$ : Final OFF

Q6:  $\text{OFF} \xrightarrow{1} \text{ON} \xrightarrow{2} \text{OFF} \xrightarrow{3} \text{ON}$ : Final ON

### Observations

Initial State	Flip C	Final State
Q7: OFF	Even	OFF
Q10: ON	Odd	OFF
Q9: OFF	Odd	ON
Q8: ON	Even	ON

Tracing:

	0	1	2	3	4	5
Initial State	off	on	on	off	on	off
$c = 0$	0 even	1 odd	2 even	2 even	3 odd	4 even
Final State	off	off	on	off	off	off
switch	$c = c + 1$	$c = c + 1$	$c = 2$	$c = c + 1$	$c = c + 1$	$c = c + 1$
	$c = 1$	$c = 2$		$c = 3$	$c = 4$	$c = 5$

Obs1: If final state is off: We switch on bulb :  $c = c + 1$

Obs2: When will final state is off?

Initial State  $\text{bulb}[i] == \text{Flip } c \% 2$  Final State

Q7: OFF 0 = Even 0 OFF

Q10: ON 1 = Odd 1 OFF

int switches (int bulbs[]) { TC: O(N) SC: O(1)

```

int c=0, N=bulbs.length;
for(int i=0; i < N; i++) {
    if(bulb[i]==0 && c%2==0) { c=c+1 }
    else if(bulb[i]==1 && c%2==1) { c=c+1 }
}
return c;
    
```

```

int switches (int bulbs[]) { TC: O(N) SC: O(1)
int c=0, N=bulbs.length;
for(int i=0; i < N; i++) {
    if(bulb[i]==c%2) { c=c+1 }
}
return c;
    
```