Mentor - Kumar K(SDE @ Amazon ; All India Rank 12/200000 Google CodeJam 2021) [Starts at 1:45pm!]

558 students of Kumar K have cracked 10+LPA in 2023:- https://mentoring.desiqna.in

Reason for the session :- Kumar K student Anusha Yerram has cracked Atlassian SDE Offer off-campus!

Link -> https://www.desiqna.in/16089/atlassian-oa-sde1-ctc-75-lac

1<=N<=100000
1<=|s[i]|<=100000
Sum of length of all string <= 10000

Understanding :- > You are given "N" strings; you can swap a character of a particular string with any other character of some other string :-> do this operation any number of times you want

-> The final output should have a maximum number of palindromic strings.

->
N = 2
ab
bab

Output - 2 [aa,bbb]

Brilliant observation :- Any character can go anywhere.

Brilliant observation :- It is now easy to solve ; just make all the places empty and then fill those empty spots by the characters you have in such a manner that maximum strings become palindromes!

Brilliant observation :- Sort all the empty places according to their size from small to large! :-> And then start filling the smallest empty place because it has the highest chances of being a plaindrome!

Brilliant observation :- Lets handle the empty space of odd length and even length separately

Brilliant observation :- To fill even length space :-> You try all characters with even frequency and try the largest ones first greedily -> In most cases it will work -> if you exhaust all the even characters -> you will start using the odd characters :-> take them all but dont use their odd guy thats it

Brilliant observation :- Odd length? :p Ignore the middle and follow the same algorithm to fill the rest of the places as their length is even. With which char should you try to fill the middle guy ? -> Odd char frequency just take 1 from him; if it works good. If its not available take 1 frequency from any even guy

HW :-> Code for it

-> even; odd;

```
for(l=1;l<=n;l++){
   vl = length[l]
   if(vl%2==0){

      while(vl!=0 and even not empty){
         guy = even.top()
         if(guy<=vl){
            vl = vl - guy
            even.pop()
         }
         else{
            guy = guy - vl
            vl = 0
            even.push(guy)
         }
      }

      while(vl!=0 and odd not empty){
         guy = odd.top() - 1
         if(guy<=vl){
            vl = vl - guy
            odd.pop()
            odd.push(1)
         }
         else{
            guy = guy - vl
            vl = 0
            odd.pop()
```

```
            odd.push(guy)
        }

    }


  }


}
```

Last observation :-> Proof of the greedy theorem :-> Lets say you are trying to make the 4th string palindrome; and you fail do it then you stop your algorithm and declare 3 as your answer

Who is in this world told you that 5th string palindrome making wont be possible?

Lets prove it and analyse:- > Watch video for it :)