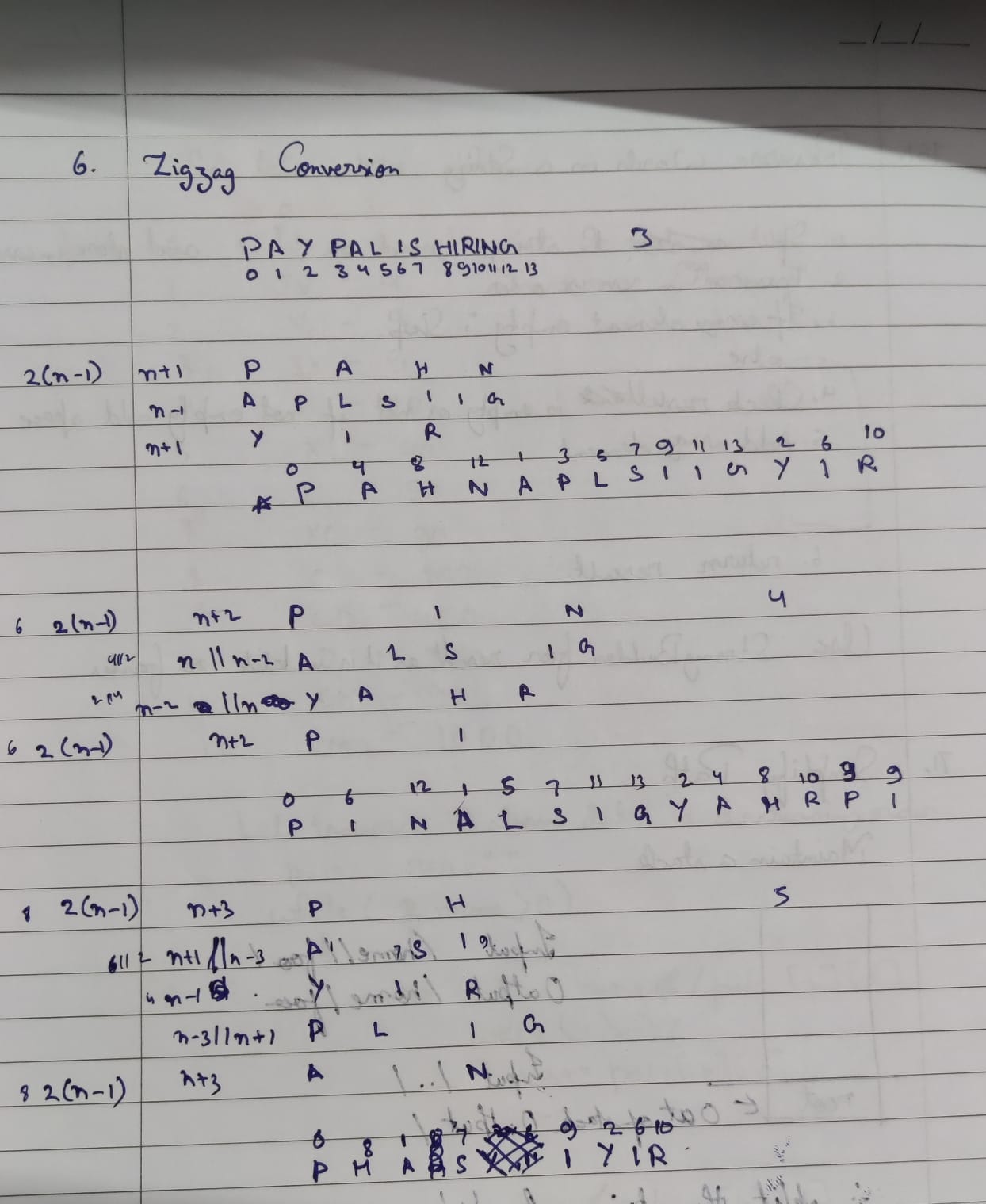
Question: https://leetcode.com/problems/zigzag-conversion/

The intution of this problem totally came into my mind after I wrote down different inputs and outputs of this problem. Once I wrote down the problem I started to figure out that at each line at what interval we were considering the elements.



Now if you have a close look when n=3.

The result’s first row has elements whose position in original string repeat after 4 indices.

when n=4.

The result’s first row has elements whose position in original string repeat after 6 indices.

when n=5.

The result’s first row has elements whose position in original string repeat after 8 indices.

Thus the interval in first and last row will be 2\*(n-1).

For the other rows similarly the increasing factor will be 2\*(n-1-(r-1)) when even pos index of that row in inserted, and for odd pos it will be 2\*(n-1)-2\*(n-1-(r-1)).

Code:  
class Solution {

public String convert(String s, int numRows) {

int l=s.length()-1;

if(l==0 || numRows==1) return s;

String res="";

for(int i=1; i<=numRows; i++){

int index=i-1, incf=2\*(numRows-1-index), z=0, uincf=0;

while(index<=l){

res+=s.charAt(index);

if(z%2==1) uincf=2\*(numRows-1)-incf;

else uincf=incf;

if(uincf==0) uincf=2\*(numRows-1);

index+=uincf;

z++;

}

}

return res;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>