Question:https://leetcode.com/problems/decode-ways/

So check if the first char is 0 or not, if 0 than there can be no possible solutions thus return 0,

else store 1 in 0th position of ways[] where we store number of ways

Now for rest of the position if the prev and present char combine to make

1. 10 or 20 - then ways[i]=ways[i-2], and if the position is 1 then ways[i]=1

2. 11-19||21-26 - then ways[i]=ways[i-1]+(i>1?ways[i-2]:1);

3. >26 - ways[i]=ways[i-2]

Code:  
class Solution {

public int numDecodings(String s) {

int[] ways= new int[s.length()];

for(int i=0; i<s.length(); i++){

if(i==0){

if(s.charAt(i)!='0')

ways[i]=1;

else return 0;

}

else{

if(s.charAt(i)=='0'){

if(s.charAt(i-1)=='1'||s.charAt(i-1)=='2'){

if(i==1)

ways[i]=1;

else

ways[i]=ways[i-2];

}else

ways[i]=0;

}else{

if(s.charAt(i-1)=='1'||(s.charAt(i-1)=='2'&&s.charAt(i)<=54)){

ways[i]=ways[i-1]+(i>1?ways[i-2]:1);

}else{

ways[i]=ways[i-1];

}

}

}

}

return ways[s.length()-1];

}

}

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