Difficulty: Medium Optimized Length: Short

Problem: Minimum Number from D/I Sequence

Given a positive integer n and a string s consisting only of letters D or I, you have to find any permutation of first n positive integer that satisfy the given input string.
D means the next number is smaller, while I means the next number is greater.

Notes

1. Length of given string s will always equal to n - 1
2. Your solution should run in linear time and space.

Additional Examples sourced from Geeks for Geeks

```
Input: D        Output: 21
Input: I        Output: 12
Input: DD       Output: 321
Input: II       Output: 123
Input: DIDI     Output: 21435
Input: IIDDD    Output: 126543
Input: DDIDDIID Output: 321654798
```

Solution (Original)

```
1.  public class Solution {
2.
3.      public ArrayList<Integer> findPerm(final String A, int B) {
4.
5.          int size = B;
6.          ArrayList<Integer> result = new ArrayList(size);
7.          for(int i=0; i<size; i++){
8.              result.add(0);
9.          }
10.         int[] mode = new int[size];
11.         int len = A.length();
12.
13.         int max=1;
14.         int localMax=1;
15.         result.set(0, 1);
16.         mode[0] = 2;
17.
18.        // System.out.println("Starting the main loop: ");
19.         for(int i=0; i<len; i++){
20.             char ch = A.charAt(i);
```

```
            if( ch == 'I' ){
        //       System.out.println("\tCharacter is I at string position: "+(i+1));
                int sol_index = i+1;
                result.set(sol_index, max+1);
                mode[sol_index]=2;
          //     System.out.println("\tSet solution for index: "+(max+1));
                max+=1;
             //   System.out.println("\tSol index is: "+sol_index+" and latest max is: "+max);
               // System.out.println("");

            }
            else{
               //System.out.println("\tCharacter is D at string position: "+(i+1));
               /*<-- in this direction finds the last index that has I
                character or mode as 2 and increment it's answer*/
                int tempIndex = i;
                int sol_index = i+1;
                mode[sol_index]=1;
                while(mode[tempIndex]!=2){
                   tempIndex-=1;
                }
                /*fouund the index and incrementing the solution for this
                index*/
                //System.out.println("\tThe last index with mode I is: "+tempIndex);
                max+=1;
                result.set(tempIndex, max);
                localMax = max;
                //System.out.println("\tThe answer set for I is: "+max);
                //System.out.println("\tGetting back to original index now from i: "+tempIndex
    );
                while(tempIndex!=sol_index){
                   tempIndex+=1;
                   localMax-=1;
                 //  System.out.println("\tfor index: "+tempIndex+" ans is: "+localMax);
                   result.set(tempIndex, localMax);
                }
                //System.out.println("");

            }
        }
        return result;
    }
}
```

Difficulty: Medium Optimized Length: Short

Solution (Optimized)

```
1.  public class Solution {
2.      public ArrayList<Integer> findPerm(final String A, int B) {
3.
4.          int size = B;
5.          int largest = B;
6.          int smallest = 1;
7.          ArrayList<Integer> result = new ArrayList(size);
8.
9.          int len = A.length();
10.         for(int i=0; i<len; i++){
11.             char ch = A.charAt(i);
12.             if( ch=='I'){
13.                 result.add(smallest);
14.                 smallest+=1;
15.             }else{
16.                 result.add(largest);
17.                 largest-=1;
18.             }
19.         }
20.         result.add(smallest);
21.
22.         return result;
23.     }
24. }
```

Node:

The Streamlined version finds any possible permutation while , the above code find the smallest possible permutation hence, it’s better. However, if not asked for smallest used streamlined, it’s much shorter and elegant.

`’Anshul Aggarwal