Decode Ways

A message containing letters from `A-Z` is being encoded to numbers using the following mapping:

```
'A' -> 1
'B' -> 2
...
'Z' -> 26
```

Given an encoded message containing digits, determine the total number of ways to decode it.

For example,
Given encoded message `"12"`, it could be decoded as `"AB"` (1 2) or `"L"` (12).

The number of ways decoding `"12"` is 2.

Solution 1

```java
public class Solution {
    public int numDecodings(String s) {
        int n = s.length();
        if (n == 0) return 0;

        int[] memo = new int[n+1];
        memo[n]  = 1;
        memo[n-1] = s.charAt(n-1) != '0' ? 1 : 0;

        for (int i = n - 2; i >= 0; i--)
            if (s.charAt(i) == '0') continue;
            else memo[i] = (Integer.parseInt(s.substring(i,i+2))<=26) ? memo[i+1]
+memo[i+2] : memo[i+1];

        return memo[0];
    }
}
```

written by manky original link here

Solution 2

```
int numDecodings(string s) {
    if (!s.size() || s.front() == '0') return 0;
    // r2: decode ways of s[i-2] , r1: decode ways of s[i-1]
    int r1 = 1, r2 = 1;

    for (int i = 1; i < s.size(); i++) {
        // zero voids ways of the last because zero cannot be used separately
        if (s[i] == '0') r1 = 0;

        // possible two-digit letter, so new r1 is sum of both while new r2 is the
old r1
        if (s[i - 1] == '1' || s[i - 1] == '2' && s[i] <= '6') {
            r1 = r2 + r1;
            r2 = r1 - r2;
        }

        // one-digit letter, no new way added
        else {
            r2 = r1;
        }
    }

    return r1;
}
```

written by shichaotan original link here

Solution 3

```
 int n = s.size();
    if(n == 0 || s[0] == '0') return 0;
    if(n == 1) return 1;
    int res = 0,fn_1 = 1,fn_2 = 1;
    for(int i = 1;i < n;i++){
        int temp = fn_1;
        if(isValid(s[i])&&isValid(s[i-1],s[i]))  res+=fn_1+fn_2;
        if(!isValid(s[i])&&isValid(s[i-1],s[i])) res+=fn_2;
        if(isValid(s[i])&&!isValid(s[i-1],s[i])) res+=fn_1;
        if(!isValid(s[i])&&!isValid(s[i-1],s[i]))  return 0;
        fn_1 = res;
        fn_2 = temp;
        res = 0;
    }
    return fn_1;
}
bool isValid(char a,char b){
    return a == '1'||(a == '2' && b <='6');
}
bool isValid(char a){
    return a != '0';
}
```

written by wang.shuai.750 original link here