Sigmainfy

Learn, Share and Inspire

Home Blog About

LeetCode Decode Ways: Dynamic Programming

Overview

LeetCode Decode Ways: Dynamic Programming is used to solve decode ways problem in iterative manner, this is a modified version of finding fibonacci numbers.

LeetCode 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 and Precautions: Dynamic Programming

This could be solved by DP. Let F(n) denote the number of decode ways for a string of length n. The following is the recursive formula:

F(n) = F(n-1) + F(n-2) if s[n] is a valid encoding digit and s[n-1]s[n] is also a valid encoding number.

F(n) = F(n-1) if s[n] is a valid encoding digit and s[n-1]s[n] is NOT a valid encoding number.

F(n) = F(n-2) if s[n] is NOT a valid encoding digit and s[n-1]s[n] is a valid encoding number.

F(n) = 0   if s[n] is NOT a valid encoding digit and s[n-1]s[n] is NOT a valid encoding number.

The following Java code is accepted by the LeetCode OJ to pass this Decode Ways problem:

```
class Solution {

    public int numDecodings(String s) {

        int n = s.length();
        if (n <= 0)
            return 0;
        if (1 == n)
            return s.charAt(0) <= '9' && s.charAt(0) >= '1' ? 1 : 0;

        int a, b, c;
        int tmp;
        a = b = c = 0;
        if (s.charAt(0) <= '9' && s.charAt(0) >= '1')
           c = a = b = 1;

        for (int i = 1; i < n; ++i) {
            a = 0;
            if (s.charAt(i) <= '9' && s.charAt(i) >= '1')
                a += b;

            tmp = Integer.parseInt(s.substring(i-1, i+1));
            if (tmp >= 10 && tmp <= 26)
                a += c;

            c = b;
            b = a;
        }
        return a;
    }
}
```

Remarks:

1. The above solution is exactly the same as for solving fibnacci number but in a bit different version by conditioning whether or not to include the value of dp[i-1] or dp[i-2], while for real fibnacci number, these two values are always used to get the next number
2. 01 would not be considered as a valid encoding, so we need to test tmp against 10 and 26 rather than 1 and 26.

Summary

LeetCode Decode Ways: Dynamic Programming is used to solve decode ways problem in iterative manner, this is a modified version of finding fibonacci numbers.

Written on May 9, 2013

« LeetCode ZigZag Conversion: 2 Different Angles

LeetCode Integer to Roman: Find the Mapping »

1 Comment **sigmainfy**

♥ **Recommend** **Share**

Join the discussion…

董昊轩 • 9 months ago

i cant believe this problem resembles fibonacci number! A real life problem with fibonacci number is ar

^ | ˅ • Reply • Share ›

Technical blog to learn, share …

1 comment • a year ago•

风之凌殇 — TAT我猜lz一定是用scanf和printf做的，这题的坑应该是在如果用cin和cout，那么最后一个测试点必定超时orz

PAT 解题报告 1038. Recover the Smallest Number (30) – Sigmainfy – Technical blog …

1 comment • a year ago•

风之凌殇 —

1 comment • a year ago•

Ximin Zhang — Good article

Leetcode Sort List Analysis: Bottom Up

1 comment • a year ago•

rcgldr — The wiki article for merge sort n
algorithm for bottom up merge sort for lin
uses a small (25 to 32) …

Subscribe Add Disqus to your site Privacy