509-斐波那契数列

题述

509. 斐波那契数

难度 简单 435

斐波那契数 （通常用 `F(n)` 表示）形成的序列称为 **斐波那契数列** 。该数列由 `0` 和 `1` 开始，后面的每一项数字都是前面两项数字的和。也就是：

```
F(0) = 0，F(1) = 1
F(n) = F(n - 1) + F(n - 2)，其中 n > 1
```

给定 `n` ，请计算 `F(n)` 。

示例 1：

输入： n = 2
输出： 1
解释： F(2) = F(1) + F(0) = 1 + 0 = 1

示例 2：

输入： n = 3
输出： 2
解释： F(3) = F(2) + F(1) = 1 + 1 = 2

示例 3：

输入： n = 4
输出： 3
解释： F(4) = F(3) + F(2) = 2 + 1 = 3

思路

斐波那契数列应该很熟悉，大一上学期学C++的时候就接触过，这道题其实非常适合作为动态规划的第一道入门题，或许你不需要多想直接就能够唰唰写出代码，但是做简单题的目的是为了熟悉动态规划的五个步骤，我们把这道题做细。

动态规划

1、确定dp数组以及其下标的含义

dp[i]的定义为：第i个数的斐波那契数值是dp[i]

2、确定递推公式

为什么这是一道非常简单的入门题目呢?

因为题目已经把递推公式直接给我们了：状态转移方程 dp[i] = dp[i - 1] + dp[i - 2];

3、dp数组如何初始化

题目中把如何初始化也直接给我们了，如下：

```
dp[0] = 0;
dp[1] = 1;
```

4、确定遍历顺序

从递归公式dp[i] = dp[i - 1] + dp[i - 2];中可以看出，dp[i]是依赖 dp[i - 1] 和 dp[i - 2]，那么遍历的顺序一定是从前到后遍历的

5、举例模拟推导dp数组

按照这个递推公式dp[i] = dp[i - 1] + dp[i - 2]，我们来推导一下，当N为10的时候，dp数组应该是如下的数列：

0 1 1 2 3 5 8 13 21 34 55

如果代码写出来，发现结果不对，就把dp数组打印出来看看和我们推导的数列是不是一致的。

题解

C++

```
class Solution {
public:
    int fib(int N) {
        if (N <= 1) return N;
        vector<int> dp(N + 1);
        dp[0] = 0;
        dp[1] = 1;
        for (int i = 2; i <= N; i++) {
            dp[i] = dp[i - 1] + dp[i - 2];
        }
        return dp[N];
    }
};
```

Python3

```python
class Solution:
    def fib(self, n: int) -> int:
        # 动态规划解决斐波那契数列问题
        # fib[0] = 0          fib[1] =1
        if n < 2:
            return n
        a , b = 0 , 1
        for i in range(1,n):
            c = a + b
            a, b = b, c
        return c
```

提交结果	执行用时	内存消耗	语言	提交时间	备
通过	32 ms	15 MB	Python3	2022/04/07 12:46	
通过	36 ms	14.8 MB	Python3	2022/04/07 12:45	

思考

斐波那契数列这道题目是非常基础的题目，通过它去感受DP的解题过程。

简单题是用来掌握方法论的，动规五部曲将在接下来的动态规划讲解中发挥重要作用！

> *2022-4-7-12：47*