

# 递推基础算法讲解

大约用时： (90 mins)

下一部分：经典面试刷题环节

# 斐波那契数列的递推公式



$$F(n) = F(n - 1) + F(n - 2)$$

# 斐波那契数列的递推公式

$$F(n) = F(n - 1) + F(n - 2)$$



Leetcode-70

# 数 学 归 纳 法



**Step1:** 验证  $k_0$  成立

**Step2:** 证明如果  $k_i$  成立, 那么  $k_{i+1}$  也成立

**Step3:** 联合 Step1 与 Step2, 证明由  $k_0 \rightarrow k_n$  成立

# 如何求解递推问题



## 1、确定递推状态（学习重点）

一个函数符号  $f(x)$ ，外加这个函数符号的含义描述  
一般函数所对应的值，就是要求解的值

## 2、确定递推公式 ( $k_i \rightarrow k_{i+1}$ )

确定  $f(x)$  究竟依赖于哪些  $f(y)$  的值

## 3、分析边界条件 ( $k_0$ )

## 4、程序实现

递归 || 循环

# 墙 壁 涂 色



**【编写程序】** 给一个环形的墙壁涂颜色，颜色一共有三种，分别是红、黄、蓝，墙壁被竖直地划分成 `wallsize` 个部分，相邻的部分颜色不能相同。请你写出函数 `paintWallCounts` 计算出一共有多少种给房间上色的方案。

例如，当 `wallsize = 5` 时，下面就是一种合法方案。



由于墙壁是环形，下面的方案就不合法。



# 墙 壁 涂 色



## 1、确定递推状态

## 2、确定递推公式

例如，当  $wallsize = 5$  时，下面就是一种合法方案。



由于墙壁是环形，下面的方案就不合法。



# 墙 壁 涂 色



## 1、确定递推状态

$f[n][i][j]$  代表前  $n$  块墙壁，在不考虑头尾成环的前提下，第1块涂颜色  $i$ ，第  $n$  块图颜色  $j$  的方法总数

## 2、确定递推公式

$$f[n][i][j] = \sum_{k=0}^2 f[n-1][i][k] \quad (k \neq j)$$

例如，当 `wallsize = 5` 时，下面就是一种合法方案。



由于墙壁是环形，下面的方案就不合法。





# 墙 壁 涂 色



3、递推式的初值条件

4、如何利用  $f$  求解最终答案?

# 墙 壁 涂 色



## 3、递推式的初值条件

$$\begin{aligned} f[1] = \{ & \\ & 1, 0, 0, \\ & 0, 1, 0, \\ & 0, 0, 1 \\ & \} \end{aligned}$$

$$\begin{aligned} f[2] = \{ & \\ & 0, 1, 1, \\ & 1, 0, 1, \\ & 1, 1, 0 \\ & \} \end{aligned}$$

## 4、如何利用 $f$ 求解最终答案?

$$ans = \sum_{i=0}^2 \sum_{j=0}^2 f[wallsize][i][j] \quad (i \neq j)$$

# 墙 壁 涂 色



## 1、确定递推状态

$f[n][j]$  代表前  $n$  块墙壁，在不考虑头尾成环的前提下，第1块涂颜色 0，第  $n$  块图颜色  $j$  的方法总数

## 2、确定递推公式

$$f[n][j] = \sum_{k=0}^2 f[n-1][k] \quad (k \neq j)$$

## 3、最终答案

$$ans = 3 \times (f[n][1] + f[n][2])$$

# 墙 壁 涂 色



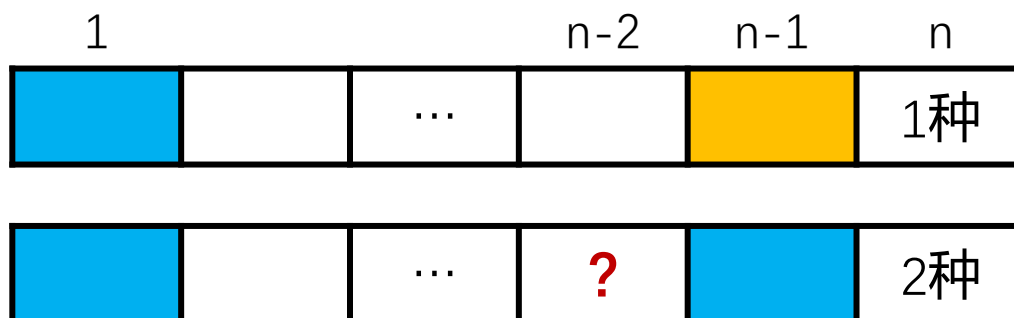
## 1、确定递推状态

$f[n]$  代表前  $n$  块墙壁，首尾颜色不同的方法总数

## 2、确定递推公式

$$f[n] = f[n-1] + 2 \times f[n-2]$$

## 4、公式的理解



# 递推与动归



# 斐波那契数列的递推公式

$$F(n) = F(n - 1) + F(n - 2)$$



Leetcode-70

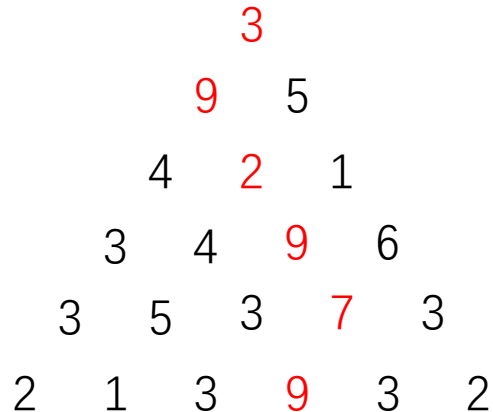


Leetcode-746/256

# Leetcode-120



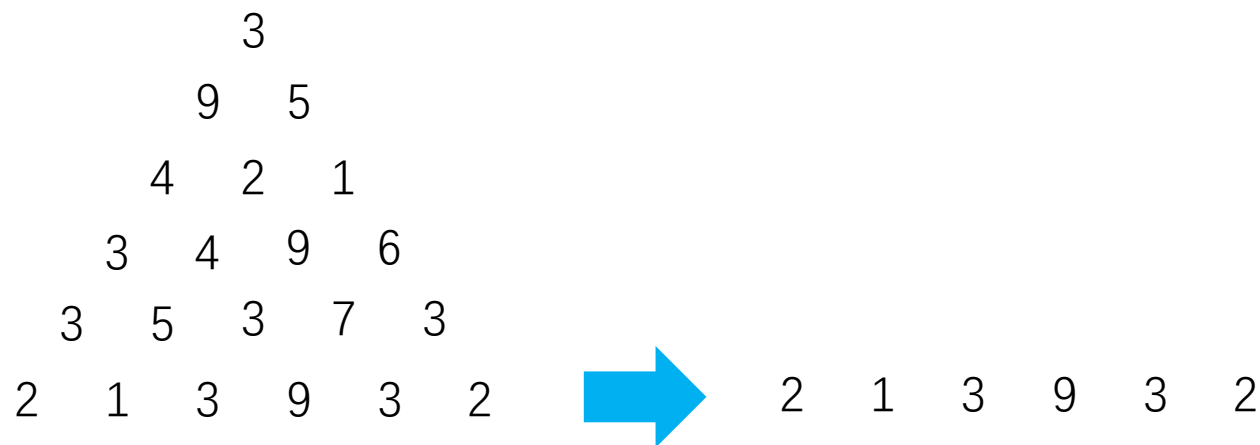
【编写程序】有一个由数字组成的三角形，站在上一层的某个点，只能到达其下方左右的两个点。现在请找到一条从上到下的路径，使得路径上所有数字相加之和最大。



# Leetcode-120



思考-1: 如果由下向上走的话, 很容易获得最后一行站在每个点上能够获得的最大值, 由此得到倒数第二行的每个点的最大值, 依次递推。

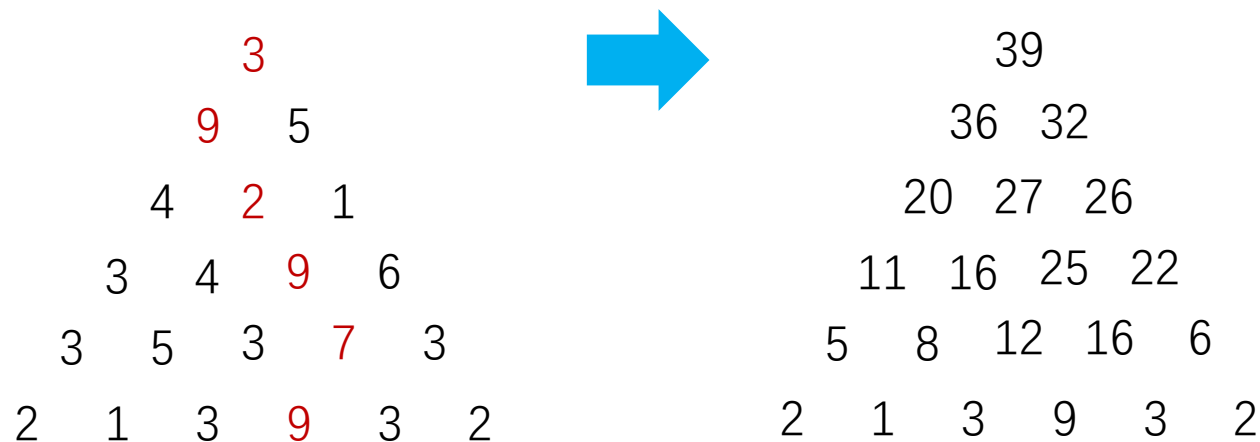




# Leetcode-120



思考-1: 如果由下向上走的话, 很容易获得最后一行站在每个点上能够获得的最大值, 由此得到倒数第二行的每个点的最大值, 依次递推。



# Leetcode-120



## 1、确定递推状态:

$f(i, j)$  代表从底边走到  $(i, j)$  点所能获得的最大值

## 2、确定递推公式

$$f(i, j) = \max[f(i+1, j), f(i+1, j+1)] + \text{val}(i, j)$$

## 3、递推公式解读

5	8	12	16	6	
2	1	3	9	3	2

Diagram illustrating the state transition for the dynamic programming problem. The top row represents the values at the current step, and the bottom row represents the values at the next step. Blue arrows point from the '3' and '2' in the bottom row to the '6' in the top row, indicating that the maximum value at the current position is derived from the maximum of the two adjacent values in the next step.

3	5	3	7	3	
2	1	3	9	3	2

# Leetcode-120



## 1、确定递推状态:

$f(i, j)$  代表从顶点走到  $(i, j)$  点所能获得的最大值

## 2、确定递推公式

$$f(i, j) = \max[f(i-1, j), f(i-1, j-1)] + \text{val}(i, j)$$

## 3、递推公式解读

22	25	26	30	18	
24	26	29	39	33	20

Diagram illustrating the grid values for the dynamic programming problem. The grid is 2 rows by 5 columns. The top row contains values 22, 25, 26, 30, 18. The bottom row contains values 24, 26, 29, 39, 33, 20. Blue arrows point from the value 30 in the top row to the value 33 in the bottom row, indicating the transition from  $f(i-1, j)$  to  $f(i, j)$ .

3	5	3	7	3	
2	1	3	9	3	2

Diagram illustrating the grid values for the dynamic programming problem. The grid is 2 rows by 5 columns. The top row contains values 3, 5, 3, 7, 3. The bottom row contains values 2, 1, 3, 9, 3, 2.

# 递 推 与 动 归



## 1、确定动归状态

例如：  $f(i, j)$  代表从底边走到  $(i, j)$  点所能获得的最大值

## 2、确定状态转移方程

$$\text{例如： } f(i, j) = \max \left\{ \begin{array}{l} f(i+1, j) \\ f(i+1, j+1) \end{array} \right\} + val(i, j)$$

## 3、正确性证明：求助于数学归纳法

## 4、程序实现

# 经典面试题刷题专项环节

大约用时：（120 mins）

下一部分：浪尖人才事业部



每天都想干翻这个世界  
到头来，被世界干的服服帖帖

大家晚安  
--船长