

279.完全平方数

题目地址: <https://leetcode-cn.com/problems/perfect-squares/>

给定正整数 n ，找到若干个完全平方数（比如 1, 4, 9, 16, ...）使得它们的和等于 n 。你需要让组成和的完全平方数的个数最少。

给你一个整数 n ，返回和为 n 的完全平方数的 最少数量。

完全平方数 是一个整数，其值等于另一个整数的平方；换句话说，其值等于一个整数自乘的积。例如，1、4、9 和 16 都是完全平方数，而 3 和 11 不是。

示例 1:

输入: $n = 12$

输出: 3

解释: $12 = 4 + 4 + 4$

1、确定dp数组及其下标

$dp[j]$:表示从下标0...i的完全平方数选取使得他们的和等于j的最少个数完全平方数

2、递推公式

$dp[j] = \min(dp[j], dp[j - \text{weight}[i]] + 1);$

3、dp数组初始化

$dp[0] = 0;$

其余初始化均为0

4遍历顺序:先遍历完全平方数再遍历组成和

```
1 class Solution {
2 public:
3     const int maxn=1000000;
4     int numSquares(int n) {
5         vector<int> dp(n+1,maxn);
6         vector<int> weight;
7         for(int i=1;i*i<=n;i++) weight.push_back(i*i);
8         dp[0]=0;
9         for(int i=0;i<weight.size();i++){
10             for(int j=weight[i];j<=n;j++){
11                 dp[j]=min(dp[j],dp[j-weight[i]]+1);
12             }
13         }
14     }
15 }
```

```
13     }  
14     return dp[n];  
15     }  
16 };  
17 //其实maxn可以用limits.h文件中的INT_MAX  
18
```

总结如下：

本周的主题其实就是背包问题中的遍历顺序！

我这里做一下总结：

求组合数： [动态规划：518.零钱兑换II](#)

求排列数： [动态规划：377. 组合总和 IV](#)、 [动态规划：70. 爬楼梯进阶版（完全背包）](#)

求最小数： [动态规划：322. 零钱兑换](#)、 [动态规划：279.完全平方数](#)

此时我们就已经把完全背包的遍历顺序研究的透透的了！