

# 0-1背包问题求解效率对比

分别使用枚举法和dp方法解决0-1背包问题，通过枚举法来校验dp方法的正确性。

问题定义：背包容量为  $V$ ，物品个数为  $n$ ，每个物品占据空间  $v_i$ ，价值  $w_i$ ，求解该背包所能容纳物品的最大价值  $W$ 。

实验要求：分别对比物品数量为4和40的情况下，使用枚举法和dp的时间。由于测试数据较少的时候，程序运行时间的波动较大，可以采用多次运行累计时间求平均值。实验所用的测试数据可以自己生成。

提交内容：

1. 一个简单的实验报告：包含时间对比的结果和相关分析即可
2. 仅包含dp算法的求解程序源代码，并能够通过以下样例

```
1 Input:
2     5 4
3     2 12
4     1 10
5     3 20
6     2 15
7 Output:
8     37
```

## 作业一：

### – 最长公共子序列问题

- 输入：第一行：随机产生序列**A**（**m**个字符）；随机产生序列**B**（**m**个字符）
- 输出：序列**A**和序列**B**的最长公共子序列（相同的机器、相同的编程语言）

问题1: 规模**m**为**5**时，穷举法和用动态规划的时间分别是多少？

问题2: 规模**m**为**25**时，穷举法和用动态规划的时间分别是多少？

# 作业二：

## **编程任务：**

现在需要你编写程序，给这  $p$  个节点安排计算任务，使得这个工程计算任务能够尽早完成。假定任务安排好后不再变动，而且所有的节点都同时开始运行，任务安排的目标是使最后结束计算的节点的完成时间尽可能早。

## **输入输出：**

输入文件名是 `hpc.in`。

文件的第一行是对计算任务的描述，包括两个正整数  $n_A$  和  $n_B$ ，分别是 A 类和 B 类子任务的数目，两个整数之间由一个空格隔开。

文件的后面部分是对此计算机的描述：

文件第二行是一个整数  $p$ ，即计算节点的数目。

随后连续的  $p$  行按顺序分别描述各个节点的信息，第  $i$  个节点由第  $i+2$  行描述，该行包括下述四个正整数（相邻两个整数之间有一个空格）： $t_i^A$   $t_i^B$   $k_i^A$   $k_i^B$

## **输出文件：**

输出文件名是 `hpc.out`。其中只有一行，包含有一个正整数，即从各节点开始计算到任务完成所用的时间。

## **样例：**

设输入文件 `hpc.in` 为

5 5

3

15 10 6 4

70 100 7 2

30 70 1 6

对应的输出文件 `hpc.out` 为

93

数据说明：

$$1 \leq n_A \leq 60, 1 \leq n_B \leq 60$$

$$1 \leq p \leq 20$$

$$1 \leq t_A \leq 1000, 1 \leq t_B \leq 1000, 1 \leq k_A \leq 50, 1 \leq k_B \leq 50$$