**0-1背包（动态规划）算法说明书**

**1. 算法功能**

给定n种物品和一背包。物品i的重量是wi，价值为vi，背包的容量为c。选择装入背包的物品，使得装入背包中物品的总价值最大。

**2. 接口参数**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 是否必须 | 示例值 | 描述 |
| 1 | ｃ | int | 是 | 4 | 背包容量 |
| 2 | w | int[] | 是 | [1, 4, 2] | 物品重量 |
| 3 | v | int[] | 是 | [4, 2, 4] | 物品价值，与重量一一对应 |

1. **接口返回值**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 名称 | 类型 | 是否必须 | 示例值 | 描述 |
| 1 | result | int[] | 是 | [1, 0, 1] | 解向量，对应物品装或不装 |

1. **算法实现**

根据矩阵连乘问题的最优值递归关系，自底向上地进行计算。在计算过程中，保存已解决的子问题的答案。每个子问题只计算一次，而在后面需要时只需简单查询，从而避免大量重复的计算，最终得到多项式时间的算法。

**5. 注意事项**

背包容量只能为整型；时间复杂度为Ｏ(nc)，物品数过大或背包容量过大都可能使得程序运行时间过长。