案例一

背包问题：有一个背包，容量为4磅 ， 现有如下物品

| **物品** | **重量** | **价格** |
| --- | --- | --- |
| 吉他(G) | 1 | 1500 |
| 音响(S) | 4 | 3000 |
| 电脑(L) | 3 | 2000 |

要求达到的目标为装入的背包的总价值最大，并且重量不超出。

解决类似的问题可以分解成一个个的小问题进行解决，假设存在背包容量大小分为1，2，3，4的各种容量的背包(分配容量的规则为最小重量的整数倍)：

例如:

| **物品** | **0 磅** | **1磅** | **2磅** | **3磅** | **4磅** |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 0 | 0 |
| 吉他(G) | 0 |  |  |  |  |
| 音响(S) | 0 |  |  |  |  |
| 电脑(L) | 0 |  |  |  |  |

v[i][0]=v[0][j]=0;

对于第一行(i=1), 目前只有吉他可以选择，所以

| **物品** | **0 磅** | **1磅** | **2磅** | **3磅** | **4磅** |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 0 | 0 |
| 吉他(G) | 0 | 1500(G) | 1500(G) | 1500(G) | 1500(G) |
| 音响(S) | 0 |  |  |  |  |
| 电脑(L) | 0 |  |  |  |  |

对于第二行(i=2),目前存在吉他和音响可以选择,所以

| **物品** | **0 磅** | **1磅** | **2磅** | **3磅** | **4磅** |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 0 | 0 |
| 吉他(G) | 0 | 1500(G) | 1500(G) | 1500(G) | 1500(G) |
| 音响(S) | 0 | 1500(G) | 1500(G) | 1500(G) | 3000(S) |
| 电脑(L) | 0 |  |  |  |  |

对于第三行(i=3),目前存在吉他和音响、电脑可以选择,所以

| **物品** | **0 磅** | **1磅** | **2磅** | **3磅** | **4磅** |
| --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 0 | 0 |
| 吉他(G) | 0 | 1500(G) | 1500(G) | 1500(G) | 1500(G) |
| 音响(S) | 0 | 1500(G) | 1500(G) | 1500(G) | 3000(S) |
| 电脑(L) | 0 | 1500(G) | 1500(G) | 2000(L) | 3500(L+G) |

w[i] 和 j 之间的关系 推导..

**案例一**：v[1][1] = ? w[1] = 1 j = 1

使用公式：

v[i][j]=max{v[i-1][j],v[i-1][j-w[i]]+v[i]} = v[1][1] = max{v[0][0], v[0][0]+v[1]} = max{0, 0 + 1500}

**案例二：**v[3][4] = ? i = 3 j = 4 w[3] = 3

j >= w[3]

使用公式：

v[i][j]=max{v[i-1][j],v[i-1][j-w[i]]+v[i]} = v[3][4] = max{v[2][4], v[2][1]+v[3]} = max{3000, 1500 + 2000} = max{3000, 3500} = 3500