**算法设计与分析**

多维背包问题就是维数不定的背包问题。再简化点说就是一件物品多种限制。比如0-1背包里物品有价值和重量的限制，那么多维背包就可能出现物品有价值、重量、购买价格、对此物品喜好程度等一系列限制。

**1、背包问题：**有m件物品和一个承重为t的背包。第i件物品的重量是w[i]，价值是v[i]。求解将哪些物品装入背包可使这些物品的重量总和不超过背包承重量t，且价值总和最大。

**2、变量对应**

* m ---count
* t --- max\_weight
* w[i] --- weight[i]
* v[i] --- value[i]

**3、子问题定义状态；**即f[i][t]表示前i件物品恰放入一个承重为t的背包可以获得的最大价值。则其状态方程式： f[i][t]=max[f[i-1][t-w[i]]+v[i],f[i-1][t];也可以写作：f[t]=max(f[t],f[t-w[i]]+v[i]);

**4、检验不同物品放入时背包总价值伪代码**

for(i = 1; i <= m; i++)

for(j = t; j >= w[i]; j--)

total\_value[j] = max(total\_value[j-w[i]]+v[i],total\_value[j]);

新物品重为weight[i]，为了能放进新的物品，需要让背包预留至少weight[i]的空间，total\_value[j-weight[i]]就是这个阶段背包的最大价值，放进新物品之后，再和没放新物品的状态比较，哪个价值大，就取哪个状态