

小马加编信息学教案(三十三)

背包问题 (2)

- 一. 课程内容
- 二. 知识讲解
 - 1. 多重背包
 - 2. 完全背包
- 三. 经典例题
- 四. 提高巩固

一. 课程内容

1. 多重背包
2. 完全背包

二. 知识讲解

1. 多重背包

如果01背包中的物品属性除了空间 c_i 、价值 w_i ，还有件数 k_i ，就是多重背包的模型了。简单来说，多重背包就是每件物品能使用不超过 k_i 次的背包问题。

对于多重背包问题，一个自然的想法是将可以选不超过 k_i 次的物品 i ，变为 k_i 件只能选一次的物品，然后多重背包就转化为了01背包。

可是，这样的做法会让物品的数量变得很多，效率变得很低。我们考虑二进制分组的优化方式。

回到原本的问题，我们将物品 i 拆成 k_i 件去做01背包的目的，是使得做01背包时，选择0~ k_i 件物品 i 的方案都可行。

考虑这样的事实： $0 \sim 2^k - 1$ 中的每一个数 x ，都可以表示为 $2^0, 2^1, \dots, 2^{k-1}$ 中若干个数字的和。由 x 写成二进制可知。

那么我们只需将 k_i 件物品 i 拆分成 $2^0, 2^1, \dots, 2^p, r_i$ 件物品 i 即可。

其中 $2^0 + 2^1 + \dots + 2^p + r_i = k_i, r_i < 2^{p+1}$

由上面的结论，容易证明这样的拆分也可以组合出选择0~ k_i 件物品 i 的方案。

我们将每个物品做二进制拆分后做01背包，就可以以不错的效率解决多重背包问题。多重背包问题还有一种使用单调队列的解法，超过本课程要求，可自行了解。

2. 完全背包

若01背包中每一件物品可以选用无限次，那么就是完全背包问题。

完全背包的一种解法就是把每一件物品拆成很多件，但是这样效率太差。

回想01背包的dp方式，我们的状态是前*i*种物品，占用*j*的空间。

然后从前*i*种物品到前*i+1*种物品的转移，如果决策第*i+1*种物品不选，此时的转移和01背包相同。

具体来说，仍是将*f[i][0..v]*复制给*f[i+1][0..v]*。

所以我们可以只使用一个数组实现。

再考虑选择第*i+1*种物品的决策，我们这样实现：

```
for(int i=0;i<n;i++)//从前i个的状态推出前i+1个的状态
{
    for(int j=0;j<=v-c[i+1];j++)
        if(f[j]!=-1)
            f[j+c[i+1]]=max(f[j+c[i+1]],f[j]+w[i+1]);
        //第i+1个选的转移
}
```

注意和01背包的转移有什么不同？

01背包中*j*是从*v-c[i+1]*到0的，而把循环遍历顺序颠倒，就成了完全背包。

可以这样理解，由于做转移*j*是递增的，状态*f[j]*会转移到*f[j'+c[i+1]]*，相当于选了一个物品*i+1*。然后遍历到*j*等于*j'+c[i+1]*时，会转移到*f[j+c[i+1]]*，即*f[j'+c[i+1]+c[i+1]]*相当于状态*f[j']*选了2个物品*i+1*.....如此，就可以转移出选择了任意数量物品*i+1*的方案。

注意理解消化。

三. 经典例题

1. 物品拆分

编写程序实现多重背包中的物品拆分

输入格式：

3个整数*c*, *w*, *k*, 表示一个物品占空间*c*, 价值*w*, 个数*k*

$c, w, k \leq 1000$

输出格式：

若干行，每行三个整数*c_i*, *w_i*, *k_i*, 表示拆分成一个物品占空间*c_i*, 价值*w_i*

同时满足 $c_i = c * k_i, w_i = w * k_i$

$\sum k_i = k$

且从上到下，*k_i*依次为 $2^0, 2^1, \dots, 2^p, r$

满足 $2^0 + 2^1 + \dots + 2^p + ri = ki, ri < 2^{p+1}$

样例输入	样例输出
1 2 9	1 2 1 2 4 2 4 8 4 2 4 2

2.完全背包

编程实现完全背包问题

输入格式：
第一行两个整数n，v。表示n种物品，背包容积v。
随后n行每行两个整数ci，wi，表示每一件第i种物品占空间ci，价值wi。
 $n \leq 100$
 $v, ci, wi \leq 10000$

输出格式：
一个整数表示答案

样例输入	样例输出
2 5 2 2 5 1	4

四. 提高巩固

1. 多重背包

一个背包，承量有限为W，有n种物体，第i种物体，价值Vi，占用重量为 Wi，且有Ci件，选择物品若干放入背包，使得总重量不超过背包的承重。总价值最大？

输入格式：
第1行，2个整数，N和W中间用空格隔开。
N为物品的种类，W为背包的容量。(1 <= N <= 100, 1 <= W <= 50000)
第2 - N + 1行，每行3个整数，Wi, Pi和Ci分别是物品体积、价值和数量。
(1 <= Wi, Pi <= 10000, 1 <= Ci <= 200)

输出格式：
输出可以容纳的最大价值。

样例输入	样例输出
3 6 2 2 5 3 3 8 1 4 1	9

2. 换零钱

n元钱有多少种方法换成零钱？
零钱有1、2、5分,1、2、5角,1、2、5、10、20、50、100元共13种面额。

输入格式：
一个整数n
 $n \leq 10000$

输出格式：
一个整数表示方案数。对1000000007取余数。

样例输入	样例输出
5	4