小马加编信息学教案(三十二)

背包问题（1）

一. 课程内容

1. 了解背包问题
2. 01背包

二. 知识讲解

1. 了解背包问题

> 背包问题是一个动态规划中的一个模型。
> 由于它太过经典，所以我们有学习的必要。

> 背包问题的模型是问题中有一个“背包”（容积限制），若干“物品”。物品有体积，可能还会有价值等属性。现在要把物品放进背包，在不超过背包容积的情况下，追求最大价值和或者其他目标。

> 根据不同问题的条件，背包问题大致可以分为01背包、无限背包、多重背包三种基本模型以及其它变种问题。

2. 01背包

> 在01背包的问题中，物品拥有价值，每个物品都只有一件，求取在不超过背包容量限制的情况下能够放进背包的物品最大价值和。

> 01背包是最经典的背包问题。考虑这样一个动态规划:
> 令f[i][j]表示前i个物品中，选取的物品体积总和为j时能获得的最大价值。
> 为什么要这样设计状态呢?
> 因为每个物品只能使用一次，所以从1到n依次考虑。
> 同时，对于任意一个前i个物品中的选取方案，我们只关心总体积和总价值。
> 又对于同样的总体积我们只关心最大总价值，于是这样设计状态顺理成章。

> 再考虑转移:
> 对于一个可达到的状态f[i][j]，设第i+1个物品体积a[i+1],价值b[i+1]。
> 现在在状态f[i][j]下我们队第i+1个物品做决策，有选和不选两种选择。
>
> 不选:
> 状态变为前i+1个物品，总体积不变，总价值不变-->用f[i][j]更新f[i+1][j]
>
> 选:
> 状态变为前i+1个物品，总体积变为j+a[i+1],总价值变为f[i][j]+b[i+1]
> -->用f[i][j]+b[i+1]更新f[i+1][j+a[i+1]]

> 初始状态f[0][0]=0
> 令m表示背包容积，做完dp后max{ f[n][0...m] }就是答案

三. 经典例题

1. 01背包问题

> n件物品和容量为v的背包，第i件物品空间ci，价值wi。
> 问不超过容量限制可装进背包最大价值和。
> $n \leq 100$
> $v \leq 10^6$
> $ci, wi \leq 10^4$

> 输入格式:
> 第一行两个整数n，v
> 第二行n个整数ci
> 第三行n个整数wi

> 输出格式:
> 一个整数表示答案

样例输入	样例输出
4 20 8 9 5 2 5 6 7 3	16

代码解析

> 这种模型题目，最重要的是熟悉和理解代码。
> 下面来看一下题目的核心部分代码。

```
void dp()
{
    memset(f,-1,sizeof(f));//假设所有状态都不可达到，记不可达到为-1
    f[0][0]=0;//初始状态前0个物品，总体积0可达到，最大总价值为0
    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;j++) f[i+1][j]=f[i][j];
        //第i+1个不选的转移

        for(int j=0;j<=v-c[i+1];j++)
            if(f[i][j]!=-1)
            f[i+1][j+c[i+1]]=max(f[i+1][j+c[i+1]],f[i][j]+w[i+1]);
        //第i+1个选的转移
    }
    for(int i=0;i<=v;i++) ans=max(ans,f[n][i]);//得到答案
}
```

> 考虑一种能够节约空间的写法。
> 不难发现，第i+1个不选的转移是把f[i][0..v]复制给f[i+1][0..v]。
> 且以后f[i][0..v]再也不会被用到。
> 再者，第i+1个选的转移，我们总是从j转移到j+c[i+1]，即状态第二维递增。
> 所以,直接使用f[i][0..v]，并从后往前做这部分转移就可以得到f[i+1][0..v]。

> 体会理解下面这一段代码，它和上面的有相同的效果

```
void dp()
{
    memset(f,-1,sizeof(f));//假设所有状态都不可达到，记不可达到为-1
    f[0]=0;//初始状态前0个物品，总体积0可达到，最大总价值为0
    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=v-c[i+1];j>=0;j--)
            if(f[i][j]!=-1)
            f[i+1][j+c[i+1]]=max(f[i+1][j+c[i+1]],f[i][j]+w[i+1]);
        //第i+1个选的转移
    }
    for(int i=0;i<=v;i++) ans=max(ans,f[i]);//得到答案
}
```

> 这样，f就只用一维数组，节约了空间。

四. 提高巩固

1. 开心的金明（noip2006普及组）

> 金明今天很开心，家里购置的新房就要领钥匙了，新房里有一间他自己专用的很宽敞的房间。更让他高兴的是，妈妈昨天对他说：“你的房间需要购买哪些物品，怎么布置，你说了算，只要不超过N元钱就行”。今天一早金明就开始做预算，但是他想买的东西太多了，肯定会超过妈妈限定的N元。于是，他把每件物品规定了一个重要度，分为5等：用整数1-5表示，第5等最重要。他还从因特网上查到了每件物品的价格（都是整数元）。他希望在不超过N元（可以等于N元）的前提下，使每件物品的价格与重要度的乘积的总和最大。
>
> 设第j件物品的价格为v[j]，重要度为w[j] ，共选中了k件物品，编号依次为$j_1, j_2, ..., j_k$ ，则所求的总和为：
> $v[j_1] \times w[j_1] + v[j_2] \times w[j_2] + ... + v[j_k] \times w[j_k]$
>
> 请你帮助金明设计一个满足要求的购物单。

> 输入格式:
> 第一行，为2个正整数，用一个空格隔开：N m
> 其中N(<30000)表示总钱数，m(<25)为希望购买物品的个数。
>
> 从第2行到第m+1行，第j行给出了编号为j–1的物品的基本数据
> 每行有2个非负整数v p
> 其中v表示该物品的价格($v \leq 10000$)，p表示该物品的重要度(1-5)

> 输出格式：
> 1个正整数，为不超过总钱数的物品的价格与重要度乘积的总和的最大值

样例输入	样例输出
1000 5 800 2 400 5 300 5 400 3 200 2	3900