背包问题九讲 v2.0

崔添翼 (Tianyi Cui, a.k.a. dd_engi)

September 15, 2011

Contents

Part I
01背包问题

1 题目

有N件物品和一个容量为V的背包。放入第i件物品耗费的空间是C_i，得到的价值是W_i。求解将哪些物品装入背包可使价值总和最大。

2 基本思路

这是最基础的背包问题，特点是：每种物品仅有一件，可以选择放或不放。

用子问题定义状态：即$F[i,v]$表示前i件物品恰放入一个容量为v的背包可以获得的最大价值。则其状态转移方程便是：

$$F[i,v] = \max\{F[i-1,v], F[i-1,v-C_i] + W_i\}$$

这个方程非常重要，基本上所有跟背包相关的问题的方程都是由它衍生出来的。所以有必要将它详细解释一下："将前i件物品放入容量为v的背包中"这个子问题，若只考虑第i件物品的策略（放或不放），那么就可以转化

为一个只和前$i-1$件物品相关的问题。如果不放第i件物品，那么问题就转化为“前$i-1$件物品放入容量为v的背包中”，价值为$F[i-1,v]$；如果放第i件物品，那么问题就转化为“前$i-1$件物品放入剩下的容量为$v-C_i$的背包中”，此时能获得的最大价值就是$F[i-1,v-C_i]$再加上通过放入第i件物品获得的价值W_i。

伪代码如下：

$F[0,0..V]=0$
for i = 1 to N
  for v = C_i to V
    $F[i,v]$ = max$\{F[i-1,v], F[i-1,v-C_i]+W_i\}$

3 优化空间复杂度

以上方法的时间和空间复杂度均为$O(VN)$，其中时间复杂度应该已经不能再优化了，但空间复杂度却可以优化到$O(V)$。

先考虑上面讲的基本思路如何实现，肯定是有一个主循环$i=1..N$，每次算出来二维数组$F[i,0..V]$的所有值。那么，如果只用一个数组$F[0..V]$，能不能保证第i次循环结束后$F[v]$中表示的就是我们定义的状态$F[i,v]$呢？$F[i,v]$是由$F[i-1,v]$和$F[i-1,v-C_i]$两个子问题递推而来，能否保证在推$F[i,v]$时（也即在第i次主循环中推$F[v]$时）能够取用$F[i-1,v]$和$F[i-1,v-C_i]$的值呢？事实上，这要求在每次主循环中我们以$v=V..0$的递减顺序计算$F[v]$，这样才能保证推$F[v]$时$F[v-C_i]$保存的是状态$F[i-1,v-C_{i]}$的值。伪代码如下：

$F[0..V]=0$
for i = 1 to N
  for v = V to C_i
    $F[v]$ = max$\{F[v], F[v-C_i]+W_i\}$

其中的$F[v]=\max\{F[v],F[v-C_i]+W_i\}$一句，恰就对应于我们原来的转移方程，因为现在的$F[v-C_i]$就相当于原来的$F[i-1,v-C_i]$。如果将$v$的循环顺序从上面的逆序改成顺序的话，那么则成了$F[i,v]$由$F[i,v-C_i]$推导得到，与本题意不符。

事实上，使用一维数组解01背包的程序在后面会被多次用到，所以这里抽象出一个处理一件01背包中的物品过程，以后的代码中直接调用不加说明。

def ZeroOnePack(F,C,W)
  for v = V to C
    $F[v]$ = max$\{F[v], f[v-C]+W\}$

有了这个过程以后，01背包问题的伪代码就可以这样写：

for i = 1 to N
  ZeroOnePack(F,C_i,W_i)

4 初始化的细节问题

我们看到的求最优解的背包问题题目中，事实上有两种不太相同的问法。有的题目要求“恰好装满背包”时的最优解，有的题目则并没有要求必须把背包装满。一种区别这两种问法的实现方法是在初始化的时候有所不同。

如果是第一种问法，要求恰好装满背包，那么在初始化时除了$F[0]$为0，其它$F[1..V]$均设为$-\infty$，这样就可以保证最终得到的$F[V]$是一种恰好装满背包的最优解。

如果并没有要求必须把背包装满，而是只希望价格尽量大，初始化时应该将$F[0..V]$全部设为0。

这是为什么呢？可以这样理解：初始化的F数组事实上就是在没有任何物品可以放入背包时的合法状态。如果要求背包恰好装满，那么此时只有容量为0的背包可以在什么也不装且价值为0的情况下被“恰好装满”，其它容量的背包均没有合法的解，属于未定义的状态，应该被赋值为-∞了。如果背包并非必须被装满，那么任何容量的背包都有一个合法解“什么都不装”，这个解的价值为0，所以初始时状态的值也就全部为0了。

这个小技巧完全可以推广到其它类型的背包问题，后面也就不再对进行状态转移之前的初始化进行讲解。

5 一个常数优化

上面伪代码中的

for $i = 1$ to N
  for $v = V$ to C_i

中第二重循环的下限可以改进。它可以被优化为

for $i = 1$ to N
  for $v = V$ to $\max\{V - \Sigma_i^N W_i, C_i\}$

这个优化之所以成立的原因请读者自己思考。（提示：使用二维的转移方程思考较易。）

6 小结

01背包问题是最基本的背包问题，它包含了背包问题中设计状态、方程的最基本思想。另外，别的类型的背包问题往往也可以转换成01背包问题求解。故一定要仔细体会上面基本思路的得出方法，状态转移方程的意义，以及空间复杂度怎样被优化。