第十三章贪心法 ：109页 Q17 a)b)

110页 Q27

第十五章动态规划：用写出背包问题中解的过程（递规、元组法）

$n=5, P=[6,3,5,4,6], w=[2,2,6,5,4]$，$c=10$

17. 考虑$0\leqslant X_i\leqslant 1$而不是$X_i\in\{0,1\}$的连续背包问题。一种可行的贪心策略是：按价值密度非递减的顺序检查物品，若剩余容量能容下正在考察的物品，将其装入；否则，往背包中装入此物品的一部分。

a）对于$n=3$，$w=[100,10,10]$，$p=[20,15,15]$及$c=105$，上述装入方法获得的结果是什么？

b）证明这种贪心算法总能获得最优解。

答案:

(a)

价值密度是[0.2, 1.5, 1.5]. 则装物品的顺序是 2, 3, 1.物品 2 和 3 能被装入，装入 2 和 3 后，背包剩余容量是 85，所以物品 1 得 85％ 能被装入。所以答案是 X= [0.85, 1, 1]，总价值为 (0.85*20+15+15)=47。

(b)

考虑 $0\leqslant X_i\leqslant 1$ 而不是 $X_i\in\{0,1\}$ 的连续背包问题。假设物品已按价值密度非递减的顺序排列，$x_1 \dots x_n$ 是贪心法得到的解，$y_1 \dots y_n$ 是最优解。下面我们可以证得这两组解得到的价值总值是相等的，从而贪心法得到的解是最优的。

假设 j 是使得($x_i=y_i$, $1\leqslant i<j$，$x_j\neq y_j$)的最小下标，如果这样的 j 不存在，则两组解是同样的，因此贪心法得到的解是最优的。假设存在这样的 j，从贪心法的求解过程以及最优解是一个可行解的事实，可以推导出 $x_j > y_j$。通过减小 y_{j+1}、y_{j+2}、…，增加 y_j 的方法，可以增加 y_j 到 x_j，因为是用高价值密度的物品代替低价值密度或等价值密度的物品，所以背包总价值不可能降低。通过这种转换，得到一个新的最优解 $y_1 \dots y_n$，新的最优解与贪心法得到的解相比，如果存在 j1 使得($x_i=y_i$, $1\leqslant i<j1$，$x_{j1}\neq y_{j1}$)，那么这里的 j1 应该大于前面提到的 j。

重复做这样的转换，可以将最初的最优解转化为贪心解，并且不会降低背包的价值，因此这种贪心算法总能获得最优解。

27. 编写一个Path(p,s,i)函数，利用函数ShortestPaths计算出的p值，输出从顶点s到顶点i的一条最短路径。函数的复杂性是多少？

答：为了找到从顶点 s 到顶点 i 的一条最短路径，首先必须验证这样的路是否存在，当 p[i]≠0 时存在这样的路。当从顶点 s 到顶点 i 的存在最短路径时，可以反过来构造从 i 到 s 的路径，其顶点序列是 p[i], p[p[i]], p[p[p[i]]], ..., s。

下面给出了输出从顶点 i 到顶点 s 的最短路径的代码。如果要输出从 s 到 i 的最短路径，只需用一数组保存从顶点 i 到顶点 s 的最短路径，然后从后向前输出即可。

```
void Path(int p[], int s, int i)

{
     // 输出从顶点 s 到顶点 i 的最短路径。
    if (i != s && !p[i]) {//没有从顶点 s 到顶点 i 的最短路径。
        cout << "There is no path from vertex "
               << s << " to vertex " << i << endl;
        return;
        }

    //有从顶点 s 到顶点 i 的最短路径。
    //构造并输出从顶点 i 到顶点 s 的最短路径。
    cout << "Shortest path from vertex "
           << s << " to vertex " << I
           << " is the reverse of " << i;
    while (i != s) {
        // move back one vertex
        i = p[i];
        cout << " " << i;
        }
    cout << endl;
}
```

因为最短路径中最多有 n 个顶点，所以 Path 函数的复杂性是 O(n).

3 写出背包问题中解的过程

n=5, P=[6,3,5,4,6],w=[2,2,6,5,4]，c=10

解: $f(5,y)=\begin{cases}6, & y\ge 4\\ 0, & y<4\end{cases}$ P[5]=[(0,0),(4,6)]

$$f(4,y)=\begin{cases}\max(f(5,y),f(5,y-5)+4), & y\ge w_4\\ f(5,y), & y<w_4\end{cases}$$

$$=\begin{cases}6, & 4\le y<5\\ 0, & y<4\\ 6, & 5\le y<9\\ 10, & y\ge 9\end{cases}$$

P[4]=[(0,0),(4,6),(9,10)]

$$=\begin{cases}0, & y<4\\ 6, & 4\le y<9\\ 10, & y\ge 9\end{cases}$$

$$f(3,y)=\begin{cases}\max(f(4,y),f(4,y-6)+5), & y\ge w_3\\ f(4,y), & y<w_3\end{cases}$$

$$=\begin{cases}0, & y<4\\ 6, & 4\le y<6\\ 6, & 6\le y<9\\ 10, & 9\le y<10\\ 11, & y\ge 10\end{cases}$$

p[3]=[(0,0),(4,6),(9,10),(10,11)]

$$=\begin{cases}0, & y<4\\ 6, & 4\le y<9\\ 10, & 9\le y<10\\ 11, & y\ge 10\end{cases}$$

$$f(2,y)=\begin{cases}\max(f,(3,y),f(3,y-2)+3, y\ge w_2\\ f(3,y), & y<w_2\end{cases}$$

$$=\begin{cases}0, & y<2\\ 3, & 2\le y<4\\ 6, & 4\le y<6\\ 9, & 6\le y<9\\ 10, & 9\le y<10\\ 11, & y=10\end{cases}$$

p[2]=[(0,0),(2,3),(4,6),(6,9),(9,10),(10,11)]

元组法：

1：P[5]=[(0,0),(4,6)]

2：Q＝[(5,4),(9,10)],因为(4,6)支配(5,4),所以消去(5,4)

P[4]=[(0,0),(4,6),(9,10)]

3:Q=[(6,5),(10,11)],因为(4,6)支配(6,5)，所以消去(6,5)

P[3]=[(0,0),(4,6),(9,10),(10,11)]

4:Q=[(2,3),(6,9)],
P[2]=[(0,0),(2,3),(4,6),(6,9),(9,10),(10,11)]

f(1,y)=max{f(2,y),f(2,y-2)+6}=max{f(2,10),f(2,8)+6}=max{11,9+6}=15
c=10, f(1,10)=15, f(2,10)=11,所以 x1=1,c=10-2=8
f(2,8)=9,f(3,8)=6,所以 x2=1,c=8-2=6
f(3,6)=6,f(4,6)=6,所以 x3=0,c=6
f(4,6)=6,f(5,6)=6,所以 x4=0,c=6
f(5,6)=6≠0,所以 x5=1,
所以 x=[1,1,0,0,1]