习题 2、用 1－优化法求解以下 0/1 背包问题，已知：n = 8, w = [16, 20, 4, 15, 25, 10, 5, 8]，p = [100, 200, 50, 90, 175, 50, 20, 60], c = 70。

解：贪心策略：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；否则考虑下个物品。

用贪心法求解的过程：

效益密度为[6.25, 10 , 12.5 , 6 , 7 , 5 , 4 , 7.5]，对其排序后得到的物品顺序为[3 , 2 , 8 , 5 , 1 , 4 , 6 , 7]，对应的重量为[4 , 20 , 8 , 25 , 16 , 15 , 10 , 5]。

k＝0时的计算结果为：

装入物品3、2、8、5后，背包剩余容量为13，接下来物品1、4的重量都超过了13，考察物品6，6装入后，背包剩余容量为3，考察物品7，其重量大于剩余容量，所以答案 x =（0 , 1 , 1 , 0 , 1 , 1 , 0 , 1 ），得到的效益值为535。

当k＝1时的计算结果为：

先放物品3、2、8、5，得到的效益值仍为535，贪心解同上；

先放物品1，得到效益值520，贪心解为（1，1，1，1，0，0，1，1）；先放物品4，得到的效益值为520，贪心解为（1，1，1，1，0，0，1，1）；

先放物品6，得到效益值为535，贪心解为（0，1，1，0，1，1，0，1）；先放物品7，得到效益值为505，贪心解为（0，1，1，0，1，1，1）。所以k优化法得到的解为（0 , 1 , 1 , 0 , 1 , 1 , 0 , 1 ），效益值为535。

习题 9.

(c)按任务执行时间 t_i 值从小到大排序，则所得的调度是 ACT 最小的。

(d)伪代码略。

(e)如果任务 i 排在 j 之后而 i 的执行时间 t_i 小于 j 的时间 t_j， 交换作业 i、j 的顺序可得到 ACT 值更好的解：从累计完成时间＝$nt_1+ (n-1)t_2 + \cdots +t_n$ 出发，直接计算可证明这一断言。

练习10(b).

定义 ACT＝max（ACT1，ACT2），ACT1＝第一个人的任务完成时刻，ACT2 为第二个人的任务完成时刻。

贪心策略：每个人交替地从未做的任务中选择执行时间最短的任务。初始按 t 的值从小到大对任务排序。该贪心策略不能得到优化解：对下述实例，交替分配不能做到平衡。设任务时间为(1,4,5,8)，交替分配得到 ACT＝max((1＋6),(4+12))=16。但优化分配为：任务 2、3 给第一个人，ACT1=9；其余给另一个人，ACT2＝9。

习题17.

考虑0≤X_i≤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，$x_1 \ldots x_n$ 是贪心法得到的解，$y_1 \ldots y_n$ 是最优解。下面我们可以证得这两组解得到的效益值是相等的，从而贪心法得到的解是最优的。
证明:

$x=(x_1. . . , x_n)$为贪心法产生的解;形式为

$(1, 1, . . . , x_j, 0. . . 0)$,其中 $0<x_j<1$;

$y=(y1. . . , y_n)$是优化解;

- 设 k 是 $x_i\neq y_i$ 的最小下标. 则 $k\leqslant j$ 且 $y_k<x_k$
- 将 y_k 增加到 $z_k=x_k$, 并从 $\Sigma_{i>k}y_iw_i$ 中减去$(x_k-y_k)w_k$ (无论什么方式)得到$(z_{k+1}, \cdots, z_n)$
- 证明$\Sigma_{k\leqslant i\leqslant n}z_ip_i\geqslant\Sigma_{k\leqslant i\leqslant n}y_ip_i$

这只要将$(z_i-y_i)p_i$ 改写成$(z_i-y_i)w_ip_i/w_i$ 利用 $p_k/w_k\geqslant p_i/w_i$ for $i>k$ 就可得到上述不等式.
$(x1, \cdots, z_k, z_{k+1}, \cdots, z_n)$仍是优化解，而且和贪心解的前 k 项相等。重复上述步骤，最终得到一个优化解，它就是贪心解。

22. （a）算法的伪代码如下:

```
      将节点按其度数从大到小排序，排序得到: v1,v2,…,vn;
 for j←1 to n do
{ S←{vj};
 for i←1 to n do
 {
 检查S∪{vi}是否构成完全子图;
 如构成完全子图，将vi加入到S中，
 否则舍弃vi;
 设该集团的节点数为dj;
 }/*找出包含vj的集团*/
 }
 从dj中找出最大者，输出对应的集团;
```

(b)例如对书中图 13.12(a)中的图，上述算法输出最大集团;反例也很容易找到。