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。

22．（a）将节点按其度数从大到小排序，排序得到:v1, v2, …, vn;

初始：S 包含节点 v1；i=2;

循环 for i：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;

(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。

20．(1)将顶点 1 加入到 S；检查顶点 1 覆盖的 B 中的顶点；

(2)重新计算 A 中除顶点 1 之外其余顶点覆盖 B 中的未被覆盖的顶点数；

(3)选出最大新的覆盖数最大的顶点，并重复(1)(2)。