1. 假定我们不再一直选择最早结束的活动，而是选择最晚开始的活动，前提仍然是与之前选出的所有活动均兼容。描述如何利用这一方法设计贪心算法，并证明算法会产生最优解。
算法描述

假设所有活动已经按**开始**时间**逆序**排序

```
GREEDY-ACTIVITY-INVERSE-SELECTOR(s, f)
n=s. length
A={a1}
k=1
for m=2 to n
    if f[m] ≤ s[k]
        A=A∪{am}
        k=m
return A
```

证明：

记活动集 $S=\{a_1,a_2,a_3\cdots\cdots a_n\}$，$a_i=\{s_i, f_i\}$，假设活动开始时间最晚的是 a_m，那 a_m 必然在其中一个最大兼容解中。因为如果存在最优解 $S'=\{a_{i1},a_{i2},a_{i3}\cdots\cdots a_{ik}\}$，$a_m$ 不在 S'中，S'中必然存在一个开始时间最晚的活动 a_k，用 a_m 替换 a_k，S'仍然兼容。

从另一个角度来想，记活动集 $S'=\{a'_1,a'_2,a'_3\cdots\cdots a'_n\}$，其中 $a'_i=\{f_i, s_i\}$，子集 $B'=\{a'_{i1},a'_{i2},a'_{i3}\cdots\cdots a'_{ik}\}\subseteq S'$兼容当且仅当 $B=\{a_{i1},a_{i2},a_{i3}\cdots\cdots a_{ik}\}\subseteq S$ 兼容，因此 S'的最优解与 S 的最优解一一映射。在 S'中使用按结束最早时间排序的 GREEDY-ACTIVITY –SELECTOR 算法求最优解，得到的结果映射到 S 中就是按开始最晚时间排序的结果。

2. 设计算法，在 O(n)时间内求解分数背包问题。

记第 i 件商品的价值为 V_i，重量为 W_i，背包的承重为 W。

(1) 找到商品单位价值 v/m 的中位数 m

(2) 将所有物品分为三类$\mathrm{G}=\left\{\frac{\mathrm{Vi}}{\mathrm{Wi}}>\mathrm{m}\right\}, \mathrm{E}=\left\{\frac{\mathrm{Vi}}{\mathrm{Wi}}=\mathrm{m}\right\}, \mathrm{L}=\left\{\frac{\mathrm{Vi}}{\mathrm{Wi}}<\mathrm{m}\right\}$，计算三类的总重量$\mathrm{W}_G=\sum_{i\in G} w_i$，$\mathrm{W}_E=\sum_{i\in E} w_i$，$\mathrm{W}_L=\sum_{i\in L} w_i$

(3) 讨论以下情况：

  a) $\mathrm{W}_G>\mathrm{W}$，不放任何物品，从(1)开始对G的物品，W重量分类

  b) 否则将W_G中所有物品放入背包中

    i. 如果$\mathrm{W}_G+\mathrm{W}_E\geq W$，从$E$中拿出$W-\mathrm{W}_G$的物品把背包放满

    ii. 把G,E的物品都放入背包，从(1)开始对L的物品，$W-\mathrm{W}_G-\mathrm{W}_E$重量分类

寻找中位数的线性查找算法伪代码为：

```
MIDDLE_PARTITION (V, W, start, end)
i = start
for j=i+1 to end
    if V[j]/W[j] ≥ V[start]/W[start] //左边放大的，右边放小的
        i = i + 1
        EXCHANGE (V, W, i, j)
EXCHANGE (V, W, i, start)
```

```
len = end-start+1
if i < len/2
    return MIDDLE_PARTITION(V, W, i+1, end)
else if i > len/2
    return MIDDLE_PARTITION(V, W, start, i-1)
else
    return V[i]/W[i]
```

求解分数背包问题的伪代码为:

```
PACKAGE_SELECT (V, W, start, end, weight)
    if start == end
        return V[start]
    middle = MIDDLE_PARTITION (V, W, start, end, V.length)
    w=0, v=0
    for i = start to middle
        w = w + W[i]
        v = v + V[i]
    if w == weight
        return v
    else if w < weight && w+W[middle] ⩾ weight
        return v + V[middle]/W[middle] * (weight-w)
    else if w < weight
        return PACKAGE_SELECT(V,W,middle+1,end,weight-w-W[middle])
    else
        return PACKAGE_SELECT(V,W,start,middle-1,weight)
```

3. 推广霍夫曼算法，使之能生成三进制的码字，并证明你的算法能生成最优三进制码。

算法伪码:

```
TRI-HUFFMAN(C)
n=|C|
Q=C
for i=1 to n-1
    allocate a new node k
    k.first = x = EXTRACT-MIN(Q)
    k.second = y = EXTRACT-MIN(Q)
    k.third = z = EXTRACT-MIN(Q)
    INSERT(Q, k)
return EXTRACT-MIN(Q)
```

证明:

(1) 首先证明，令 x, y, z 是 C 中频率最低的三个字符，那么存在 C 的一个最优前缀码，x 和 y 和 z 的码字长度相同，且只有最后一个进制位不同。

令 T 表示任意一个最优前缀码对应的编码树，a，b，c 是 T 中深度最大的兄弟叶节点。a.freq≤b.freq≤c.freq，x.freq≤y.freq≤z.freq，因为 x,y,z 为 C 中频率最低的三个字符，所以有 x.freq≤a.freq，y.freq≤b.freq，z.freq≤c.freq。在 T 中交换 x 和 a 生成一棵新树 T'，在 T'

中交换 b 和 y 生成一棵新树 T''，在 T'''中交换 c 和 z 生成一棵新树 T'''。T 和 T'的代价差为：

$$\begin{aligned}\mathrm{B(T)}-\mathrm{B(T')}&=\sum_{c\in C} c.freq\cdot d_T(c)-\sum_{c\in C} c.freq\cdot d_{T'}(c)\\&=\mathrm{x.freq}\cdot d_T(x)+a.freq\cdot d_T(a)-\mathrm{x.freq}\cdot d_{T'}(x)-a.freq\cdot d_{T'}(a)\\&=\mathrm{x.freq}\cdot d_T(x)+a.freq\cdot d_T(a)-\mathrm{x.freq}\cdot d_T(a)-a.freq\cdot d_T(x)\\&=(a.freq-x.freq)(d_T(a)-d_T(x))\geq 0\end{aligned}$$

同理，$\mathrm{B(T')}-\mathrm{B(T'')}\geq 0$，$\mathrm{B(T'')}-\mathrm{B(T''')}\geq 0$，因此 T'''也是最优树，且 x,y,z 是其中最深的兄弟节点。

(2)接着证明，令 C'为 C 去掉 x 和 y 和 z，加入一个新字符 k 后得到的字母表，k.freq=x.freq+y.freq+z.freq，T'为字母表 C'的任意一个最优前缀码对应的编码树，我们可以将 T'中叶结点 k 替换为以 x,y,z 为孩子的内部结点，得到树 T，而 T 就是表示字母表 C 的一个最优前缀码。

由于$\mathrm{d}_T(x)=\mathrm{d}_T(y)=\mathrm{d}_T(z)=\mathrm{d}_{T'}(k)+1$，可以得到：

$$\begin{aligned}\mathrm{x.freq}\cdot\mathrm{d}_T(x)+\mathrm{y.freq}\cdot\mathrm{d}_T(y)+\mathrm{z.freq}\cdot\mathrm{d}_T(z)&=(\mathrm{x.freq}+\mathrm{y.freq}+\mathrm{z.freq})(\mathrm{d}_{T'}(k)+1)\\&=\mathrm{k.freq}\cdot\mathrm{d}_{T'}(k)+(\mathrm{x.freq}+\mathrm{y.freq}+\mathrm{z.freq})\end{aligned}$$

因此可以得到结论

$$\mathrm{B(T)}=\mathrm{B(T')}+\mathrm{x.freq}+\mathrm{y.freq}+\mathrm{z.freq}$$
$$\mathrm{B(T')}=\mathrm{B(T)}-(\mathrm{x.freq}+\mathrm{y.freq}+\mathrm{z.freq})$$

假定 T 对应的前缀码不是 C 的最优前缀码。存在最优编码树 T''满足 B(T'')<B(T)，T''包含兄弟结点 x,y,z，令 T'''为 T''中将 x,y,z 以及他们的父结点替换为 k 得到的树，可以得知

$$\begin{aligned}\mathrm{B(T''')}&=\mathrm{B(T'')}-\mathrm{x.freq}-\mathrm{y.freq}-\mathrm{z.freq}\\&<\mathrm{B(T)}-\mathrm{x.freq}-\mathrm{y.freq}-\mathrm{z.freq}=\mathrm{B(T')}\end{aligned}$$

这与 T'对应 C'的一个最优编码树相矛盾，因此 T 是 C 的一个最优前缀码。

由(1)和(2)可知，TRI-HUFFMAN 会生成一个最优前缀码。