16.1-2

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。

解：

将输入活动按照开始时间单调递减排序，每次选择与之前选出的活动均兼容且最晚开始的活动。

参考定理16.1的证明，证明算法会产生最优解即证明：考虑任意非空子问题S_k，令a_m是S_k中开始时间最晚的活动，则a_m在S_k的某个最大兼容活动子集中。

证明：令 A_k 是 S_k 的一个最大兼容活动子集，且 a_j 是 A_k 中开始时间最晚的活动。若 $a_j = a_m$，则已经证明 a_m 在 S_k 的某个最大兼容活动子集中。若 $a_j \neq a_m$，令集合$A'_k = A_k - \{a_j\} \cup \{a_m\}$，即将 A_k 中的 a_j 替换为 a_m 。 A'_k 中的活动都是不相交的，因为 A_k 中的活动都是不相交的， a_j 是 A_k 中开始时间最晚的活动，而 $s_m \geqslant s_j$ 。由于 $|A'_k| = |A_k|$，因此得出结论 A'_k 也是 S_k 的一个最大兼容活动子集，且它包含 a_m 。

16.1-3

对于活动选择问题，并不是所有贪心方法都能得到最大兼容活动子集。请举例说明，在剩余兼容活动中选择持续时间最短者不能得到最大集。类似地，说明在剩余兼容活动中选择与其他剩余活动重叠最少者，以及选择最早开始者均不能得到最优解。

解：

选择持续时间最短的反例

选择与其他剩余活动重叠最少的反例

选择最早开始的反例

16.3-2

证明：一棵不满的二叉树不可能对应一个最优前缀码。

解：

若存在只有一个子结点(S)的结点(T)，直接用S替代T，可以获得更优的前缀码。

16.3-3

如下所示，8 个字符对应的出现频率是斐波那契数列的前 8 个数，此频率集合的赫夫曼编码是怎样的？

a：1　b：1　c：2　d：3　e：5　f：8　g：13　h：21

你能否推广你的结论，求频率集为前 n 个斐波那契数的最优前缀码？

解：

(1)

a: 1111111

b: 1111110

c: 111110

d: 11110

e: 1110

f: 110

g: 10

h: 0

(2)

$$\begin{aligned}
&\operatorname{code}(c_n)=0\\
&\operatorname{code}(c_{i-1})=1\operatorname{code}(c_i)\qquad 2\leq i\leq n-1\\
&\operatorname{code}(c_1)=1^{n-1}
\end{aligned}$$

23.2-1

对于同一个输入图，Kruskal 算法返回的最小生成树可以不同。这种不同来源于对边进行排序时，对权重相同的边进行的不同处理。证明：对于图 G 的每棵最小生成树 T，都存在一种办法来对 G 的边进行排序，使得 Kruskal 算法所返回的最小生成树就是 T。

解：

先将边按权重排序。在挑选边的时候，如果出现多条边权值相等，先挑在T中包含的边。

23.2-2

假定我们用邻接矩阵来表示图 $G=(V, E)$。请给出 Prim 算法的一种简单实现，使其运行时间为 $O(V^2)$。

解：

```
PRIM-ADJ(G, w, r)
    initialize A with every entry = (NIL, ∞)
    T = {r}
    for i = 1 to V
        if Adj[r, i] != ∞
            A[i] = (r, Adj[r, i])
    for each u in V - T
        // 找到剩余点到集合T最近的点，加入集合
        k = min(A[i].2)
        T = T U {k}
        k.π = A[k].1
        // 更新剩余点到集合T的距离
        for i = 1 to V
            if Adj[k, i] != ∞ and Adj[k, i] < A[i].2
                A[i] = (k, Adj[k, i])
```

24.3-1

在图 24-2 上运行 Dijkstra 算法，第一次使用结点 s 作为源结点，第二次使用结点 z 作为源结点。以类似于图 24-6 的风格，给出每次 **while** 循环后的 d 值和 π 值，以及集合 S 中的所有结点。

解：

以 **s** 为源结点

d 值：

s	t	x	y	z
0	3	∞	5	∞
0	3	9	5	∞
0	3	9	5	11
0	3	9	5	11
0	3	9	5	11

π 值:

s	t	x	y	z
NIL	s	NIL	s	NIL
NIL	s	t	s	NIL
NIL	s	t	s	y
NIL	s	t	s	y
NIL	s	t	s	y

以 **z** 为源结点

d 值:

s	t	x	y	z
3	∞	7	∞	0
3	6	7	8	0
3	6	7	8	0
3	6	7	8	0
3	6	7	8	0

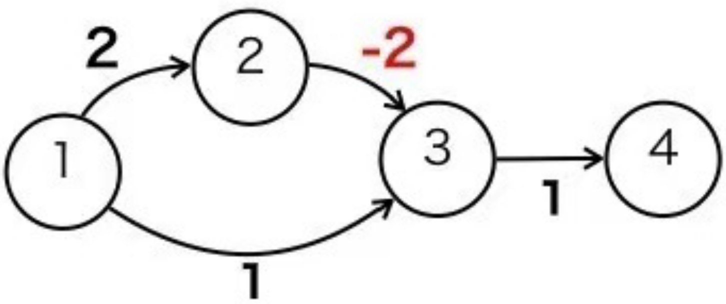
π 值:

s	t	x	y	z
z	NIL	z	NIL	NIL
z	s	z	s	NIL
z	s	z	s	NIL
z	s	z	s	NIL
z	s	z	s	NIL

24.3-2

请举出一个包含负权重的有向图，使得 Dijkstra 算法在其上运行时将产生不正确的结果。为什么在有负权重的情况下，定理 24.6 的证明不能成立呢?

解:



Dijkstra算法正确的前提是长的路径是由短的路径派生而来的。但是一旦存在了负的权值，那么一个最短路径就可能在增加这个负权值的路线后变成更短的路径。（定理24.6的证明中$\delta(s,\ y) \leq \delta(s,\ u)$不成立）

24.3-4

Gaedel 教授写了一个程序，他声称该程序实现了 Dijkstra 算法。对于每个结点 $v \in V$，该程序生成值 $v.d$ 和 $v.\pi$。请给出一个时间复杂度为 $O(V+E)$ 的算法来检查教授所编写程序的输出。该算法应该判断每个结点的 d 和 π 属性是否与某棵最短路径树中的信息匹配。这里可以假设所有的边权重皆为非负值。

解：

1.先检验s.d=0和s.π=NIL

2.遍历每一个结点u，检验是否u.d = u.π.d + w(u.π, u)

3.遍历每一个结点u，对其邻接结点v进行松弛操作，如果松弛操作后v.d改变则错误

```
For each u ∈ V-1
     For each v ∈ G.Adj[u]
           temp = v.d
           RELAX(u, v, w)
           If v.d != temp
                 return False
return True
```

时间复杂度： O(1)+O(V)+O(V+E)

29.1-4

将下面线性规划转换成标准型：
最小化 $$2x_1 + 7x_2 + x_3$$
满足约束
$$\begin{aligned} x_1 \qquad\quad - x_3 &= 7 \\ 3x_1 + x_2 \qquad\quad &\geqslant 24 \\ x_2 \qquad\quad &\geqslant 0 \\ x_3 &\leqslant 0 \end{aligned}$$

解：

$$\begin{array}{lrcrcrcrl} \text{maximize} & -2x_1 & - & 2x_2 & - & 7x_3 & + & x_4 & \\ \text{subject to} & & & & & & & & \\ & -x_1 & + & x_2 & - & x_4 & \leq & -7 & \\ & x_1 & - & x_2 & + & x_4 & \leq & 7 & \\ & -3x_1 & + & 3x_2 & - & x_3 & \leq & -24 & \\ & & & x_1, x_2, x_3, x_4 & & & \geq & 0 & \end{array}$$

29.1-5

将下面线性规划转换成松弛型：
最大化 $$2x_1 - 6x_3$$
满足约束
$$\begin{array}{rcrcrcl} x_1 & + & x_2 & - & x_3 & \leqslant & 7 \\ 3x_1 & - & x_2 & & & \geqslant & 8 \\ -x_1 & + & 2x_2 & + & 3x_3 & \geqslant & 0 \\ & & x_1, x_2, x_3 & & & \geqslant & 0 \end{array}$$
其中基本变量和非基本变量是什么？

解：

$$\begin{array}{rclcrcrcr} x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -8 & + & 3x_1 & - & x_2 & & \\ x_6 & = & -x_1 & + & 2x_2 & + & 2x_3 & & \\ x_1, x_2, x_3, x_4, x_5, x_6 & \geq & 0 & & & & & & \end{array}$$

基本变量：x_4，x_5，x_6

非基本变量：x_1，x_2，x_3

29.1-6

说明下面线性规划是不可解的：

最大化 $3x_1-2x_2$

满足约束

$$\begin{aligned} x_1 + x_2 &\leqslant 2 \\ -2x_1 - 2x_2 &\leqslant -10 \\ x_1, x_2 &\geqslant 0 \end{aligned}$$

解：

约束条件$x_1+x_2 \le 2$和$-2x_1-2x_2 \le -10$相互矛盾

29.3-5

采用 SIMPLEX 求解下面的线性规划：

最大化 $18x_1+12.5x_2$

满足约束

$$\begin{aligned} x_1 + x_2 &\leqslant 20 \\ x_1 &\leqslant 12 \\ x_2 &\leqslant 16 \\ x_1, x_2 &\geqslant 0 \end{aligned}$$

解：

$$\text{maximize} \quad 18x_1 + 12.5x_2$$

$$\begin{aligned} \text{subject to} \quad x_3 &= 20 - x_1 - x_2 \\ x_4 &= 12 - x_1 \\ x_5 &= 16 - x_2 \\ x_1, x_2, x_3, x_4, x_5 &\geq 0. \end{aligned}$$

最优解为$(12, 8, 0, 0, 8)$，目标值为316