离散数学: 作业一

刘泓尊　2018011446　计 84

1. 证明. 以工厂为节点，工厂之间有联系则建边，构成无向图。

a. 假设每座工厂都只与其他 3 座工厂有联系，则总度数为

$$\sum_{v \in V(G)} d(v) = 9 \times 3 = 27$$

由性质 1.1.1，图的度数和应为偶数，矛盾！

故，不可能每座工厂都只与其他三座工厂有业务联系。

b. 假设只有 4 座工厂与偶数个工厂有业务联系，剩余 5 个工厂与奇数个工厂有联系，则总度数为

$$\sum_{v \in V(G)} d(v) = 4 \times 2i + 5 \times (2j+1) = 2(4i+5j) + 1$$

上式为奇数，同样与性质 1.1.1 矛盾！

故，不可能只有 4 座工厂与偶数个工厂有联系. □

2. 证明. 假设 G 中存在至少 1 个孤立节点，则至多有 $n-1$ 个非孤立节点。由于 G 为简单图，不存在重边和自环，所以每个非孤立节点最多与 $n-2$ 个非孤立节点相连，总边数

$$m = \frac{1}{2}\sum_{v \in V(G)} d(v) \le \frac{1}{2}\sum_{v \in V(G)} (n-2) \le \frac{1}{2}(n-1)(n-2)$$

与题设条件

$$m > \frac{1}{2}(n-1)(n-2)$$

矛盾!

故，G 中不存在非孤立节点。 □

3. 证明. 根据定义，有向完全图满足条件

$$d^+(v_i) + d^-(v_i) = n-1, \quad v_i \in V(G)$$

故

$$\begin{aligned} left - right &= \sum_{v_i \in V(G)} \left((d^+(v_i))^2 - (d^-(v_i))^2\right) \\ &= \sum_{v_i \in V(G)} \left((d^+(v_i))^2 - (n-1-d^+(v_i))^2\right) \\ &= \sum_{v_i \in V(G)} \left(2(n-1)d^+(v_i) - (n-1)^2\right) \\ &= 2(n-1)\sum_{v_i \in V(G)} d^+(v_i) - n(n-1)^2 \\ &= 2(n-1)\frac{n(n-1)}{2} - n(n-1)^2 \\ &= 0 \end{aligned} \tag{1}$$

即

$$\sum_{v_i \in V(G)} (d^+(v_i))^2 = \sum_{v_i \in V(G)} (d^-(v_i))^2$$

□

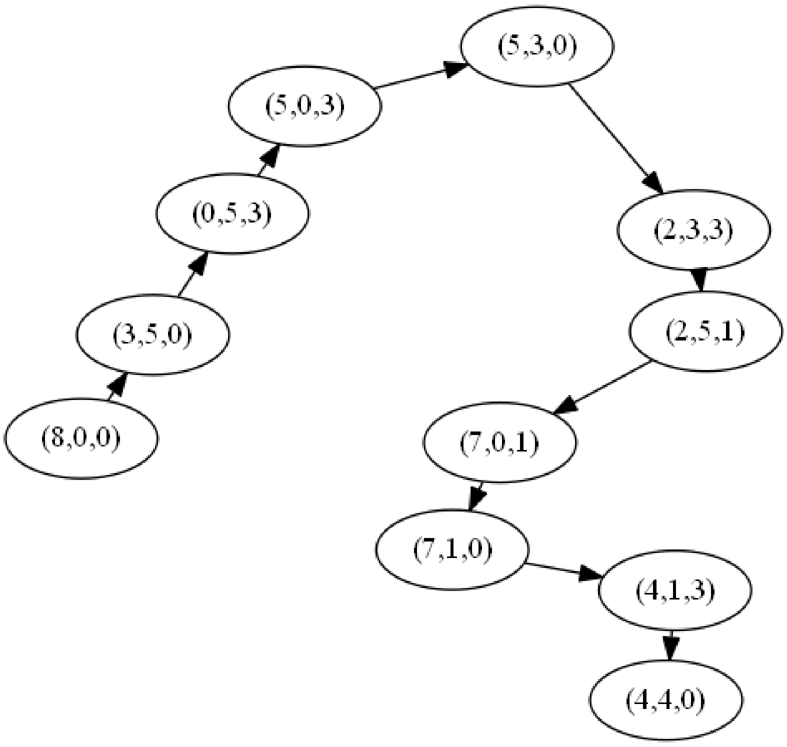
4. 设 $8L, 5L, 3L$ 的容器对应节点 (a, b, c), 其中 a, b, c 分别依次对应三种容器。

搜索思路:

对状态进行转移与搜索 (dfs)，当某一容器 m 非空且存在另一容器 n 不满时，就可以将水从 m 倒入 n 中，使得 m 空或者 n 满，以此作为一次状态转移;

对各状态进行 DFS 之后可以得到如下图所示的解。解的路径就是操作的具体过程。

- 附录 (Appendix) 给出了具体实现代码。



7. 同构。构造映射 $f: G_a \to G_b$ 如下:

v	v_1	v_2	v_3	v_4	v_5	v_6
$f(v)$	b	a	c	e	d	f

8.

邻接矩阵：

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

设边的顺序为 $e_1 = (v_1, v_2), e_2 = (v_1, v_4), e_3 = (v_3, v_1), e_4 = (v_2, v_5), e_5 = (v_6, v_3), e_6 = (v_6, v_4), e_7 = (v_5, v_3), e_8 = (v_3, v_4), e_9 = (v_6, v_1)$

关联矩阵：

$$\begin{pmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

边列表：

$$A = (1, 1, 3, 2, 6, 6, 5, 3, 6)$$

$$B = (2, 4, 1, 5, 3, 4, 3, 4, 1)$$

正向表：

$$A = (1, 3, 4, 6, 6, 7, 10)$$

$$B = (2, 4, 5, 1, 4, 3, 3, 4, 1)$$

Thinking. 完全二分图 $K_{m,n}$ 何时为欧拉图?

由于无向连通图 G 存在欧拉回路 (是欧拉图) 的充要条件是 G 中每个节点的度均为偶数。

而完全二分图两子集 X, Y 中，X 中节点度为 $|Y|$,Y 中节点度为 $|X|$, 故充要条件为：

$$m, n \text{ 均为偶数}$$

2. 证明. 假设 G 和 $\overline{G}$ 均不是连通图

设 G 存在连通支 G_1, G_2, 在 G 中任取节点 v_1, v_2:

• 若存在边 (v_1, v_2), 则在 G_2 中取点 v_3,v_3 与 v_1, v_2 均不连通，由补图的定义，$\overline{G}$ 中 v_3 与 v_1, v_2 连通，则 $\overline{G}$ 中 v_1 与 v_2 通过 v_3 连通。

• 若不存在边 (v_1, v_2), 则由补图的定义，$\overline{G}$ 中 v_1 与 v_2 通过 v_3 连通.

由 v_1, v_2 的任意性知，$\overline{G}$ 连通。与假设矛盾！

故 G 和 $\overline{G}$ 至少有一个是连通图。 □

3. 证明. 图 G 有两条最长道路 L_1, L_2, 其长度均为 l, 假设两条道路不相交，即两者没有公共顶点

由于 G 为连通图，L_1, L_2 中分别存在点 v_1, v_2 使得存在 $v_1 \to v_2$ 的道路，设该道路为 L. v_1 分 L_1 为两部分，一部分 L_{1a} 长度 $\geq l/2$; 同理，v_2 分 L_2 为两部分，一部分 L_{2a} 长度 $\geq l/2$，则取道路 $L_{1a} \cup L_{2a} \cup L$，该道路长度 $\geq l/2 + l/2 + 1 = l + 1$, 与 L_1, L_2 是最长道路的假设矛盾！

故若连通图最长道路不唯一，它们必定相交。 □

4. 证明. 对顶点数 n 作归纳。

a.$n = 4$ 时，$m \geq 5$。

若 G 中存在度为 0 或 1 的节点，则剩余 3 个节点的子图的度 $\sum deg \geq 4$，与简单图的边数矛盾。故 G 中节点 $deg \geq 2$. 则 G 中回路只有两种形式:

1. 三角形回路　2. 四边形回路

• 若为三角形回路，则 G 中剩下的一个顶点至少与此三角形有 2 条边，这两条边与原三角形两边构成四边形回路，三角形剩下的一条边就是弦 ($Chord$).

• 若为正方形回路，则至少还有一条边，该边就是弦。

b. 假设对节点数 $n-1$ 的图成立.

c. 对于节点数为 n 的图:

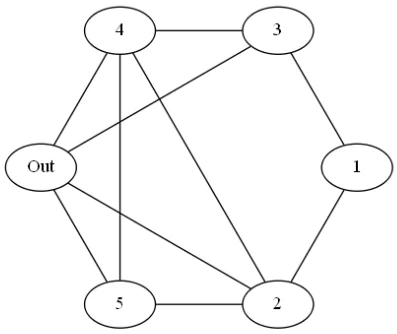
• 若 G 中存在度为 1 的节点 v，则将该节点及其所连边去除后，由归纳假设知命题成立.

• 若 G 中存在度为 2 的节点 v，设该节点所连点为 v_s, v_t, 则去除 v 及其所连边，并将 v_s, v_t 合并为一个节点，该图即为 $n-1$ 的图 G'，由归纳假设知 G' 存在带弦回路，还原为 G 后仍然是带弦回路。

• 否则，G 中所有节点 $deg \geq 3$，由例 2.1.3 结论可知，G 中必含带弦回路。 □

6. 解. 将图中房间设为顶点，同时增设外部节点 Out, 两节点之间边代表有门；构造新图 G 如下.

问题转化为 G 中是否存在欧拉道路，由于 G 中存在度为奇的两个点 v_3, v_5, 由推论 2.3.1 得 G 中存在欧拉道路，所以存在一条路经过每个门一次。 □



8. 证明. 假设 G 不存在 H 回路，则由推论 2.4.1，存在节点 v_i, v_j 使得 $d(v_i)+d(v_j)<n$.
考虑图 $G' = G - v_i, v_j$, 即图 G 删去节点 v_i, v_j 及其所连边的情况.
一方面，

$$m_{G'} = m - (d(v_i)+d(v_j)) > m - n \geq \frac{1}{2}(n-1)(n-2) + 2 - n = \frac{1}{2}(n-2)(n-3)$$

另一方面，

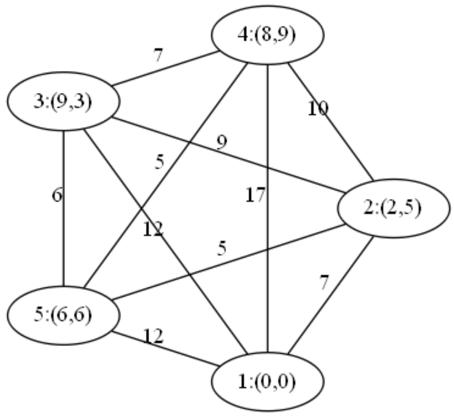
$$m_{G'} < m_{K_{n-2}} = \frac{1}{2}(n-2)(n-3)$$

上述两式矛盾，所以 G 中存在 H 回路。 □

12. 证明. 不能.
将正方体的 27 个节点建模为二分图 (或 2-染色)，具体做法为:
8 个顶点和 6 个面的中心点归为点集 X，12 条棱中点和体中心点归为点集 Y, 那么相邻节点之间必为一个 X, 一个 Y。
而一个角的顶点为 X，中心为 Y, 根据二分图的性质，从点集 X 走到点集 Y 中所经过路径长度必然为奇数；而遍历 27 个顶点仅一次只需要 26 步，是偶数，矛盾！
所以不能实现. □

13. 解. 将 4 个坐标点连同原点建图，以坐标为节点，坐标之间的曼哈顿距离为边权，构建完全图 K_5, 如下图所示
问题转化为求下图的最短哈密顿回路，即 TSP，采用分支与界法可以求得问题的解。
1. 将边权递增排序，初始界 $d_0 \leftarrow \infty$

e_{ij}	a_{25}	a_{45}	a_{35}	a_{12}	a_{34}	a_{23}	a_{24}	a_{13}	a_{15}	a_{14}
w_{ij}	5	5	6	7	7	9	10	12	12	17



2. 边权序列中依次选边进行 dfs, 直到选择 n 条边，判断是否构成 H 回路.

求解过程如下:

- $d(S_1) = d(a_{25}, a_{45}, a_{35}, a_{12}, a_{34})$, it's not a $Hamiltonian \quad Circle$. if a_{34} is replaced by $a_{23}, a_{24}, a_{13}, a_{15}, a_{14}$, it's still not a $Hamiltonian \quad Circle$. Likely, if a_{12} is replaced by $a_{34}, a_{23}, a_{24}, a_{13}, a_{15}$, it's still not a $Hamiltonian \quad Circle$.
- $d(S_2) = d(a_{25}, a_{45}, a_{12}, a_{34}, a_{23})$, it's not a $Hamiltonian \quad Circle$. Until $d(S_3) = d(a_{25}, a_{45}, a_{12}, a_{34}, a_{13}) = 36$, it's a $Hamiltonian \quad Circle$, $d_0 \leftarrow 36$.
- $d(S_4) = d(a_{25}, a_{45}, a_{34}, ..., ...)$ are all not a $Hamiltonian \quad Circle$.
- $d(S_5) = d(a_{25}, a_{35}, a_{12}, a_{34}, a_{14}) = 42$, it's a $Hamiltonian \quad Circle$.

类似的，继续进行下去得到的解会比 36 大，所以 TSP 问题的解是

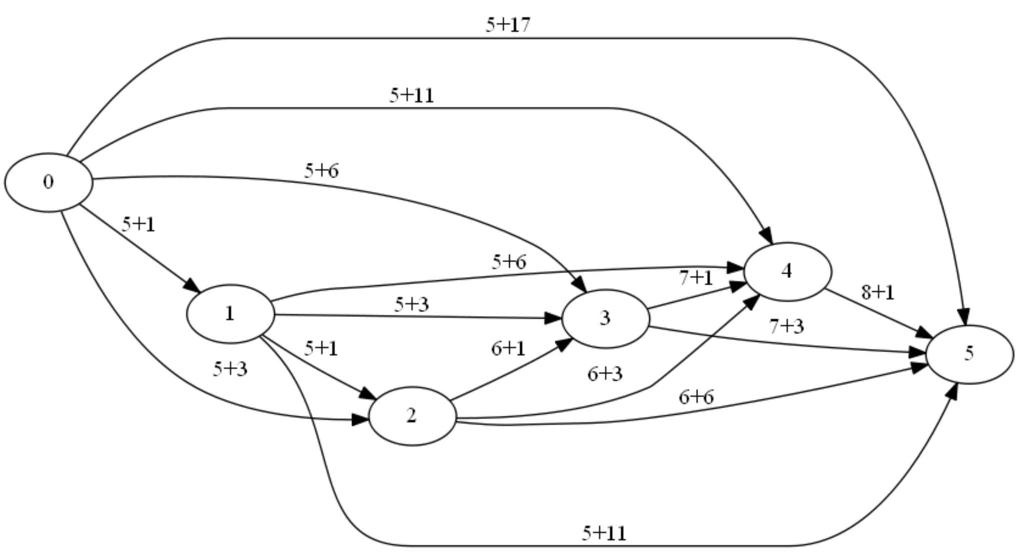
$$d(S_3) = d(a_{25}, a_{45}, a_{12}, a_{34}, a_{13}) = 36$$

即行进路线为:

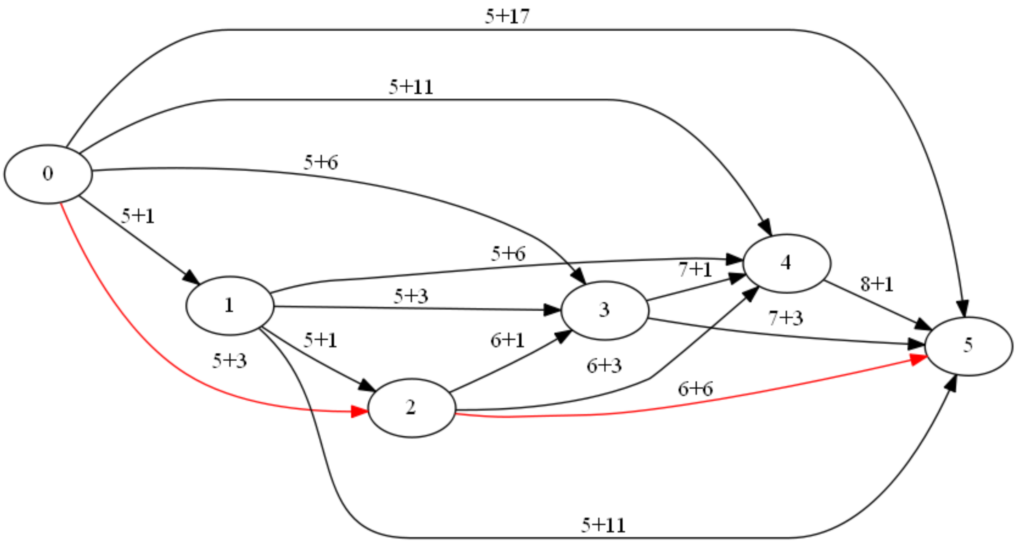
$$(0,0) \to (2,5) \to (6,6) \to (8,9) \to (9,3) \to (0,0)$$

最短路径长 36. □

15. 解. 可以将 5 年的 6 个时间点看做节点，节点之间的有向边代表 1 台设备从一年用到另一年的总花费，形成 DAG，最小花费即为该图从起点到终点的最短路径，使用最短路算法可以解决这个问题。1 台设备累计花费为费用数组的前缀和，即 $(1, 3, 6, 11, 17)$, 建图如下:



对上图使用最短路算法，得到的最短路如下:



所以最少开支的方案为:

第 1 年购买 1 台设备使用 2 年 (花费 5+1+2 = 8)，第 3 年购买 1 台设备使用 3 年 (花费 6+1+2+3 = 12)。总开支 20。 □

16. 解. 采用 *Edmonds* 最小权匹配求解中国邮路。

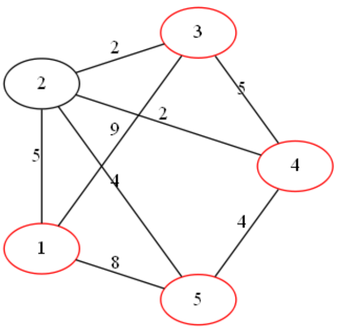


图 2: deg 为奇

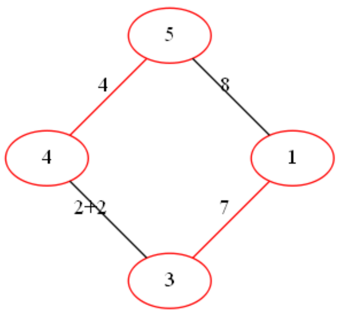


图 3: 最小权匹配

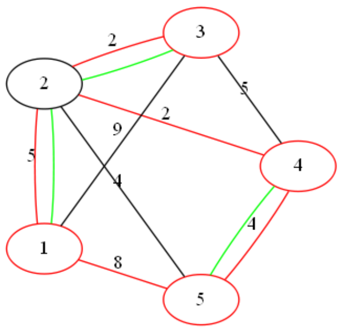
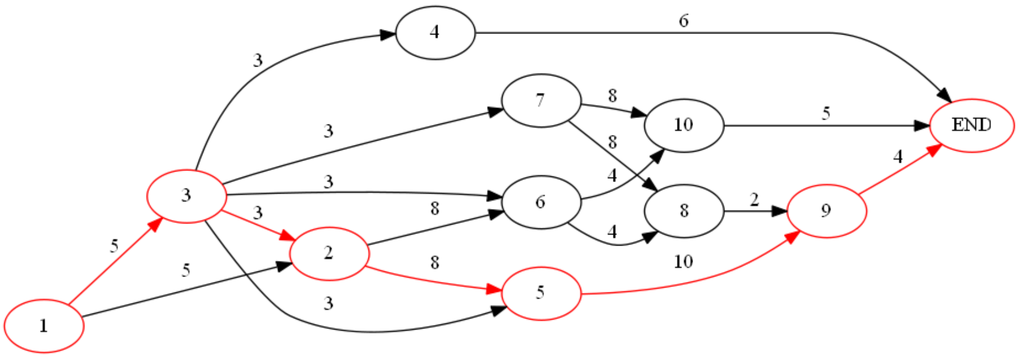


图 4: 中国邮路

1. 找到图中度为奇的节点 (图中红色节点)。
2. 对图中红色节点寻找在原图中的最短路，进行最小权匹配 (图中红色边为选中的边)。
3. 将最小权匹配中选择的边设置为重边 (图中绿色边)，得到欧拉图 G'.
4. G' 的一条欧拉回路就是解。

Conclusion: 中国邮路长度为 $d = 50$, 重复走的边为 $1-2, 2-3, 4-5$. □

17. 解. 工序的 PT 图如下图所示:



对上图使用拓扑排序算法，得到工序的拓扑序为:

TopoRank	1	2	3	4	5	6	7	8	9	10	11
Vertex	1	3	7	4	2	6	10	8	5	9	END

关键路径如上图红色路径所示。
得到工序 3,5,10 的最早启动时间为

$$\pi(3) = 5, \pi(5) = 16, \pi(10) = 20$$

最晚启动时间为

$$\tau(3) = 5, \tau(5) = 16, \tau(10) = 25$$

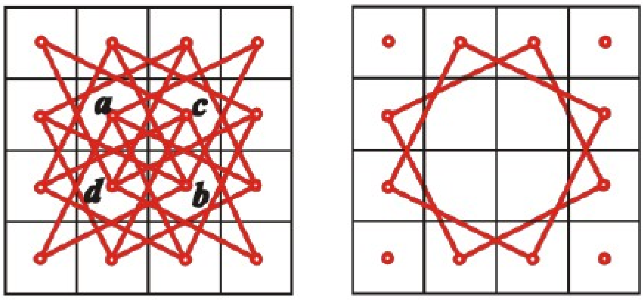
允许延误时间为

$$t_3 = 0, t_5 = 0, t_{10} = 5$$

□

Thinking. $\frac{1}{4}$ 国际象棋棋盘跳马问题?

证明. 原问题可转化为求解下图的哈密顿回路问题:



如图，删去 $S = \{a, b, c, d\}$ 四个点，得到 $G - S$. 因为

$$p(G - S) = 6 > |S| = 4$$

故该图不存在哈密顿回路，所以跳马问题无解。 □

1. 解. 设树有 n_1 个度为 1 的节点，总节点数为 n，$k \geq 3$ 时，由树的性质:

$$n_1 + 2n_2 + 3n_3 + \cdots + kn_k = 2(n-1)$$

$$n_1 + n_2 + n_3 + \cdots + n_k = n$$

联立以上两式，解得:

$$n_1 = 2 + n_3 + \cdots (k-2)n_k = 2 + \sum_{i=3}^{k} (i-2)n_i, \quad k \geq 3$$

$k = 2$ 时，该树只有 1 个树根和 2 个孩子，度为 1 的节点有 2 个。综上, 度为 1 的节点数:

$$n_1 = \begin{cases} 2 & k = 2 \\ 2 + \sum_{i=3}^{k} (i-2)n_i & k \geq 3 \end{cases} \tag{2}$$

□

4. 解. **a.** 原图的 $Laplacian$ 矩阵为:

$$L(G) = \begin{pmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 4 & -1 & 0 & -2 \\ 0 & -1 & 4 & -2 & -1 \\ -1 & 0 & -2 & 4 & -1 \\ -1 & -2 & -1 & -1 & 5 \end{pmatrix}$$

树的数目为:

$$det(L') = det \begin{pmatrix} 3 & -1 & 0 & -1 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -2 \\ -1 & 0 & -2 & 4 \end{pmatrix} = 101$$

b.$G/\{v_1, v_5\}$(缩点) 的 $Laplacian$ 矩阵:

$$L(G/\{v_1, v_5\}) = \begin{pmatrix} 6 & -3 & -1 & -2 \\ -3 & 4 & -1 & 0 \\ -1 & -1 & 4 & -2 \\ -2 & 0 & -2 & 4 \end{pmatrix}$$

必含 (v_1, v_5) 的树的数目:

$$det(L') = det \begin{pmatrix} 6 & -3 & -1 \\ -3 & 4 & -1 \\ -1 & -1 & 4 \end{pmatrix} = 44$$

c. $G - (v_1, v_5)$(删边) 的 $Laplacian$ 矩阵:

$$L(G - (v_1, v_5)) = \begin{pmatrix} 3 & -1 & 0 & -1 & -1 \\ -1 & 4 & -1 & 0 & -2 \\ 0 & -1 & 4 & -2 & -1 \\ -1 & 0 & -2 & 3 & 0 \\ -1 & -2 & -1 & 0 & 4 \end{pmatrix}$$

不含 (v_4, v_5) 的树的数目:

$$det(L') = det \begin{pmatrix} 3 & -1 & 0 & -1 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -2 \\ -1 & 0 & -2 & 3 \end{pmatrix} = 60$$

□

5. 解. **a.** 原图的 $Laplacian$ 矩阵为:

$$L(G) = \begin{pmatrix} 0 & -1 & 0 & -1 & -1 \\ 0 & 2 & -1 & 0 & -1 \\ 0 & 0 & 3 & -1 & 0 \\ 0 & 0 & -1 & 3 & 0 \\ 0 & -1 & -1 & -1 & 2 \end{pmatrix}$$

以 v_1 为根的根树的数目为:

$$det(L') = det\begin{pmatrix} 2 & -1 & 0 & -1 \\ 0 & 3 & -1 & 0 \\ 0 & -1 & 3 & 0 \\ -1 & -1 & -1 & 2 \end{pmatrix} = 24$$

b.$G-(v_1,v_5)$(删边) 的 *Laplacian* 矩阵:

$$L(G-(v_1,v_5)) = \begin{pmatrix} 0 & -1 & 0 & -1 & 0 \\ 0 & 2 & -1 & 0 & -1 \\ 0 & 0 & 3 & -1 & 0 \\ 0 & 0 & -1 & 3 & 0 \\ 0 & -1 & -1 & -1 & 1 \end{pmatrix}$$

以 v_1 为根，不含 (v_1,v_5) 的根树的数目:

$$det(L') = det\begin{pmatrix} 2 & -1 & 0 & -1 \\ 0 & 3 & -1 & 0 \\ 0 & -1 & 3 & 0 \\ -1 & -1 & -1 & 1 \end{pmatrix} = 8$$

c. $G-(v_2,v_3)$(删边) 的 *Laplacian* 矩阵:

$$L(G-(v_2,v_3)) = \begin{pmatrix} 0 & -1 & 0 & -1 & -1 \\ 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 3 & 0 \\ 0 & -1 & -1 & -1 & 2 \end{pmatrix}$$

不含 (v_2,v_3) 的树的数目:

$$det(L') = det\begin{pmatrix} 2 & 0 & 0 & -1 \\ 0 & 2 & -1 & 0 \\ 0 & -1 & 3 & 0 \\ -1 & -1 & -1 & 2 \end{pmatrix} = 15$$

必含 (v_2,v_3) 的，以 v_1 为根的根树的数目

$$24-15=9$$

□

8. 证明. 完全二分图 $K_{m,n}$ 的 *Laplacian* 矩阵:

$$L(G) = \begin{pmatrix} nI_m & -B \\ -B^T & mI_n \end{pmatrix}$$

其中 B 为 $(m \times n)$ 的全 1 矩阵。

树的数目为:

$$
\begin{aligned}
det(L_{m+n,m+n}) &= det\,(nI_m) \cdot det\left(mI_n - B^T \frac{1}{n} I_{n-1} B\right) \\
&= n^m \cdot det \begin{pmatrix} m-\frac{m}{n} & -\frac{m}{n} & \cdots & -\frac{m}{n} \\ -\frac{m}{n} & m-\frac{m}{n} & \cdots & -\frac{m}{n} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{m}{n} & -\frac{m}{n} & \cdots & m-\frac{m}{n} \end{pmatrix} \\
&= n^m \cdot \frac{m^{n-1}}{n} \\
&= n^{m-1} m^{n-1}
\end{aligned} \tag{3}
$$

□

10. 解. **1.**

$$
B = (B_{11}, B_{12}) = \left(\begin{array}{cccc|cccc} -1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 \end{array}\right)
$$

$$
B_{12}^{-1} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 \end{pmatrix}
$$

以 $\{e_3, e_4, e_6, e_7\}$ 为树边的基本回路矩阵:

$$
C_f = (I, -B_{11}^T B_{12}^{-T}) = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & -1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{pmatrix}
$$

2.

$$
B = (B_{11}, B_{12}) = \left(\begin{array}{cccc|cccc} -1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 \end{array}\right)
$$

$$
B_{12}^{-1} = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}
$$

以 $\{e_2, e_5, e_6, e_8\}$ 基本割集矩阵:

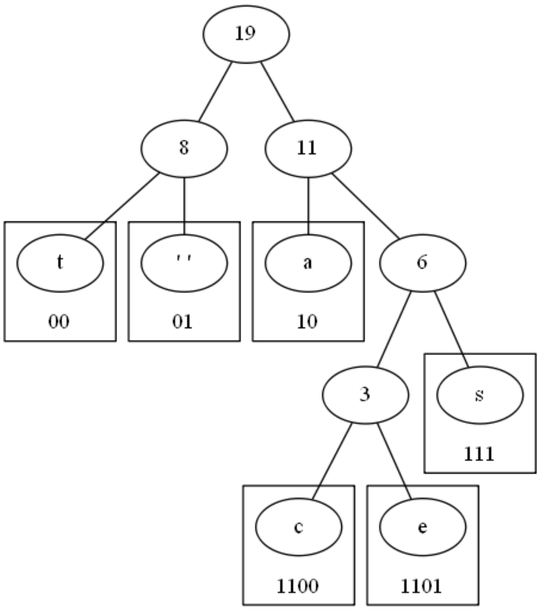
$$
S_f = (B_{12}^{-1} B_{11}, I) = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
$$

□

14. 解. state act as a seat 的字频统计:

char	c	e	s	t	' '	a
count	1	2	3	4	4	5

$Huffman$ 树的构造及对应字符的编码如下:



二进制编码:

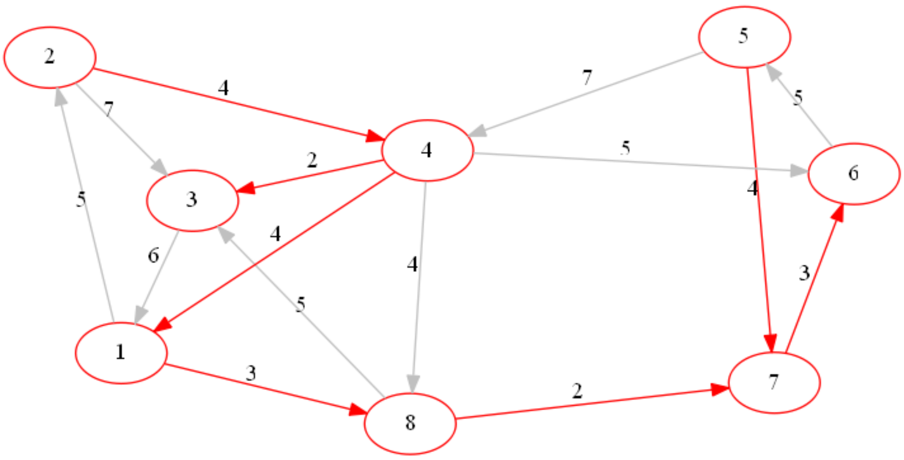
char	c	e	s	t	' '	a
code	1100	1101	111	00	01	10

得到源字符串的 $Huffman$ 编码为:

$$11100100011010110110000011011101100111111011000$$

编码长度 $codelen = 47$. □

16. 解. 对原图使用 $Kruskal$ 算法，得到的最短树如下:(树边用红色标识)



最短树树边为 $(v_1, v_8), (v_2, v_4), (v_4, v_1), (v_4, v_3), (v_5, v_7), (v_7, v_6), (v_8, v_7)$，总权 22. □

3. 证明. 假设 G 和 $\overline{G}$ 均为平面图，则

$$|E(G)| \leq 3n - 6$$

$$|E(\overline{G})| \leq 3n - 6$$

故

$$|E(G)| + |E(\overline{G})| = |E(K_n)| = \frac{n(n-1)}{2} \leq 6n - 12$$

解得

$$3 \leq n \leq 10$$

与 $n > 10$ 矛盾！
所以节点数大于 10 的简单图中，G 和 $\overline{G}$ 至少有一个是非平面图。 □

7. 证明. 假设存在这样的平面图 G 满足题设性质。
由于 G 为平面图，则其存在唯一的对偶图 G^*。根据题设，G^* 有 5 个节点，两两节点之间至少有一个边，即 G^* 存在 K_5 子图，故 G^* 不是平面图，故 G^* 不存在对偶图。
与 $(G^*)^* = G$ 矛盾！
故不存在 5 个域的平面图，每个域之间至少有一条公共边界。 □

A Appendix

test.cpp

```cpp
bool state[9][6][4] = {0};//标记此状态是否到达过
stack<Node> S;
bool dfs(int a, int b, int c){//搜索+剪枝
    state[a][b][c] = 1; S.push(Node(a, b, c));//用栈保存搜索结果
    printf("%d,%d,%d\n", a, b, c);
    if(a == 4 && b == 4 && c == 0)return true;
    if(a > 0){//对可行分支进行搜索
        if(b < 5){
            int change = min(5-b, a);
            if(!state[a-change][b+change][c] && dfs(a - change, b + change, c))return true;
        }
        if(c < 3){
            int change = min(3-c, a);
            if(!state[a-change][b][c+change] && dfs(a - change, b, c + change))return true;
        }
    }
    if(b > 0){
        if(a < 8){
            int change = min(8-a, b);
            if(!state[a+change][b-change][c] && dfs(a + change, b - change, c))return true;
```

```
        }
        if(c < 3){
            int change = min(3-c, b);
            if(!state[a][b-change][c+change] && dfs(a, b - change, c + change))return true;
        }
    }
    if(c > 0){
        if(b < 5){
            int change = min(5-b, c);
            if(!state[a][b+change][c-change] && dfs(a, b + change, c - change))return true;
        }
        if(a < 8){
            int change = min(8-a, c);
            if(!state[a+change][b][c-change] && dfs(a + change, b, c - change))return true;
        }
    }
    S.pop();
    return false;
}
int main(){
    run(8, 0, 0);
    //在此处打印栈S的内容即可
    return 0;
}
```