证明：首先证明它是一棵支撑树。采用归纳法，初始 $U=\{v_1\}$，$T=\Phi$，它是由 U 导出的树，设 $|U|=i$，T 是 U 导出的树，则下一次迭代时，U 中增加一新顶点 u，T 中也加入一条与 u 相连的边，因此 T 是连通的，有 $|U|-1$ 条边，它是由 U 导出的一棵树。因此，最终 T 是 G 的支撑树。以下再证 T 是一棵最短树。设 T_0 是 G 的一棵最短树，若 $T\neq T_0$，由定理 3.6.3，对任意的 $e\in T-T_0$，一定有最短树 $T'=T_0\oplus(e,\ e')$，其中 $e'\in C^e\cap T_0$。继续对 T' 如此处理，直至最终 $T'=T$，它仍然是最短树。

Kruskal 算法的复杂性与迭代次数有关，如果图 G 的边数很多，或称为稠密图时，p 值可能较大，也许接近 m。Prim 算法只与 G 的顶点有关，而与图的稠密度无关。因此，相比较而言，Prim 算法适用于稠密图，而 Kruskal 算法对稀疏图更为合适。

最短树问题一经解决，最长树问题也就迎刃而解。这只要将加入树的边次序按权构成非增序列，采用类似 Kruskal 算法即可实现。有兴趣的读者可以自行设计并实现最长树算法。

习　题　3

1.【★☆☆☆】一棵树有 n_2 个顶点的度为 2，n_3 个顶点的度为 3，$\cdots$，n_k 个顶点的度为 k。问有多少个度为 1 的顶点?

2.【★☆☆☆】证明：树中最长道路的两端点一定都是树叶。

3.【★☆☆☆】设 T 为一棵树，已知 T 中度数为 $1,\ 2,\ \cdots,\ i-1,\ i+1,\ \cdots,\ k$ 的点有 $a_1,\ a_2,\ \cdots,\ a_{i-1},\ a_{i+1},\ \cdots,\ a_k$ 个。其中，k 为树中度数最大的顶点的度数，且 $i\geqslant 2$。求 T 中度数为 i 的顶点数。

4.【★☆☆☆】设 T 为一棵树，任取一条边 $e=(u,\ v)\in T$，将 u 和 v 合并为一个点，原来与其相连的边连接至这个新点上，得到新图 G。证明 G 为一棵树。

5.【★☆☆☆】已知 n 个点 m 条边的无向图 G 为 k 棵树组成的森林，证明：$m=n-k$。

6.【★★☆☆】证明任意一个森林均为二分图。

7.【★☆☆☆】设 T 是顶点数为 $n\geqslant 2$ 的树，证明：对于树中的任一结点，对于以其为端点的最长初级道路，另一端点一定为树叶。

8.【★★★☆】定义树的中心为以其为端点的最长初级道路长度最小的顶点。证明：任意一棵树至多有两个中心，且当有两个中心时，这两个中心相邻。

9.【★★☆☆】设 $G=(V,\ E)$ 为有向连通图，e 是 G 的一条边。证明：

(1) 若 e 不在 G 的任何一棵支撑树中，则 e 为自环。

(2) 若 e 在 G 的每个支撑树中，则 e 为割边。

10.【★☆☆☆】求图 3.24 所示无向图中支撑树的数目。

11.【★☆☆☆】求 K_n 中不含某特定边 $(v_i,\ v_j)$ 的树的数目。

12.【★☆☆☆】求 K_n 中必含某特定边 $(v_i,\ v_j)$ 的树的数目。

13.【★★☆☆】证明：完全二分图 $K_{m,\ n}$ 的树的数目是 $m^{n-1}n^{m-1}$。

14.【★☆☆☆】求图 3.25 所示有向图中不含 $(v_1,\ v_4)$、$(v_6,\ v_5)$、$(v_6,\ v_1)$ 的支撑树的数目。

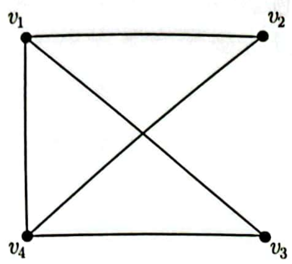

图 3.24

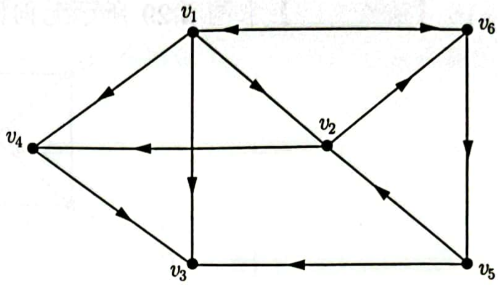

图 3.25

15.【★★☆☆】求图 3.26 所示有向图中必含 (v_1, v_2)、(v_2, v_7)、(v_4, v_6) 的支撑树的数目。

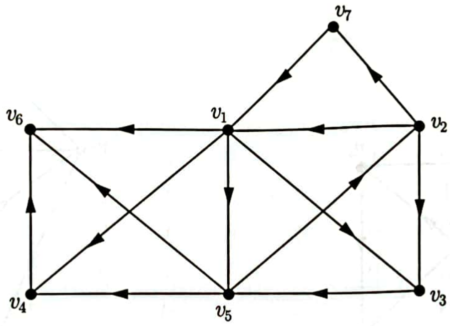

图 3.26

16.【★☆☆☆】求图 3.27 中

(1) 树的数目。

(2) 必含 (v_1, v_5) 的树的数目。

(3) 不含 (v_4, v_5) 的树的数目。

17.【★☆☆☆】求图 3.28 中

(1) 以 v_1 为根的根树的数目。

(2) 以 v_1 为根不含 (v_1, v_5) 的根树的数目。

(3) 以 v_1 为根必含 (v_2, v_3) 的根树的数目。

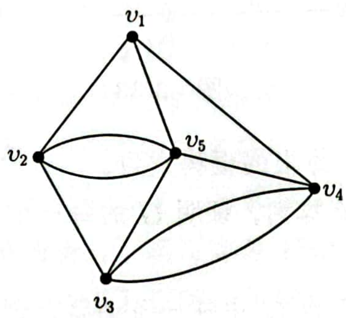

图 3.27

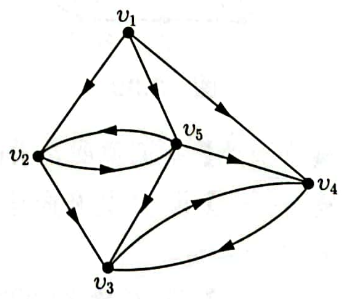

图 3.28

18.【★☆☆☆】求图 3.29 所示无向图中支撑树的数目。

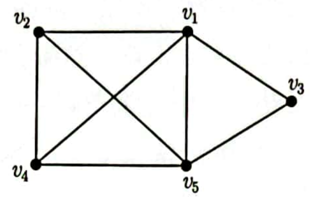

图 3.29

19.【★☆☆☆】求图 3.30 所示有向图中以 v_1 为根的根树的数目。

20.【★☆☆☆】求图 3.31 所示有向图中以 v_1 为根不含 (v_1, v_2)、(v_4, v_5) 的根树的数目。

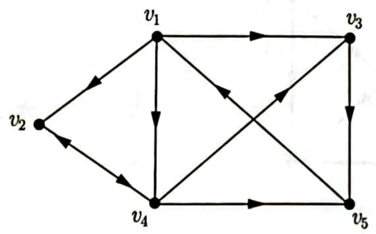

图 3.30

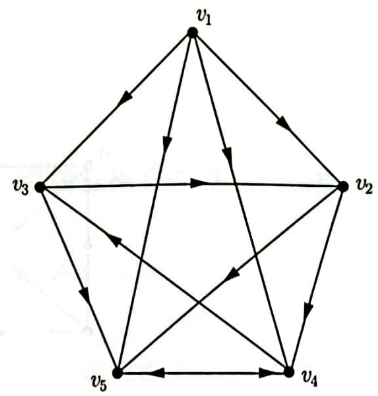

图 3.31

21.【★★☆☆】求图 3.32 所示有向图中以 v_1 为根必含 (v_2, v_4)、(v_3, v_6)、(v_6, v_5) 的根树的数目。

22.【★★☆☆】求图 3.33 所示无向图中支撑树的数目 (其中边上的数字代表这条边重复的次数)。

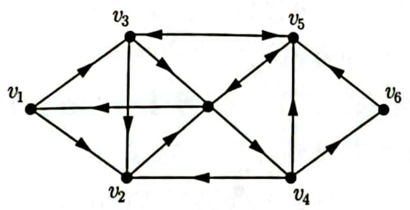

图 3.32

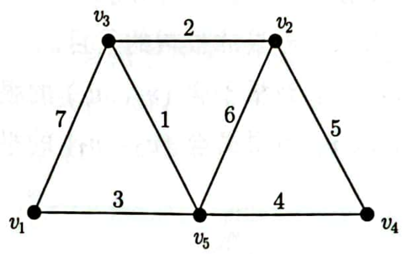

图 3.33

23.【★☆☆☆】举例说明，$\det\left(\boldsymbol{B}_k\boldsymbol{B}_k^{\mathrm{T}}\right)$ 不是以 v_k 为根的根树数目。

24.【★☆☆☆】设 T 是有向连通图 G 的任何一棵支撑树，证明 G 的每一个割集 S 都至少含有 T 的一条边。

25.【★☆☆☆】设 $\overline{T}$ 为 G 的支撑树 T 的余树，C 为 G 中任一条回路。证明：C 与 $\overline{T}$ 一定存在公共边。

26.【★★☆☆】设 T 是有向连通图 G 的一棵支撑树，e 是 $G-T$ 的一条边，C 是由 e 确定的 $T+e$ 中的基本回路，证明：e 包含在 C 中除 e 外的每条边所确定的基本割集中，而不在其他的基本割集中。

27.【★☆☆☆】已知连通图 G 的基本关联矩阵是

$$\begin{array}{cc} \boldsymbol{B}_5 = & \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 \end{bmatrix} \\ & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{matrix} \end{array}$$

求：(1) 以 $\{e_3，e_4，e_6，e_7\}$ 为树的基本回路矩阵。

(2) 以 $\{e_2，e_5，e_6，e_8\}$ 为树的基本割集矩阵。

28.【★★☆☆】已知矩阵 $\boldsymbol{C}'$ 包含了连通图 G 的回路矩阵，求 G 的以 $\{e_5，e_6，e_7，e_8\}$ 为树的基本割集矩阵。

$$\begin{array}{cc} \boldsymbol{C}' = & \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & 0 & 1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\ & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{matrix} \end{array}$$

29.【★★★☆】设 G 是无向图，对任意顶点 $v \in V(G)$，$G-v$ 仍是连通图，而且 G 的基本割集矩阵 $\boldsymbol{S}_f$ 的每一行都有偶数个 1 元素。证明 G 中有欧拉回路。

30.【★★☆☆】设完全 m 叉树中，树叶数为 t，分枝顶点数是 i，证明：$(m-1)i=t-1$。

31.【★☆☆☆】给出字符串 state act as a seat：

(1) 求最优二进制编码。

(2) 如果二进制字符串不允许带空格，求该字符串的最优二进制编码。

32.【★☆☆☆】假设数据项 A，B，C，D，E，F，G 以下面的概率分布出现：$A:0.1$，$B:0.3$，$C:0.05$，$D:0.15$，$E:0.2$，$F:0.15$，$G:0.05$，求一种二进制编码方式使得传输一个数据项的期望长度最小，并求其期望。

33.【★☆☆☆】10 个树叶的权值分别为 20、7、88、100、6、13、18、30、7、16，求构造的 Huffman 树的带权路径总长。

34.【★☆☆☆】给出字符串 abbcccddddeeeee，①求其最优二进制编码的长度；②求其最优三进制编码的长度；③求其最优四进制编码的长度。

35.【★★★☆】编写实现 Huffman 算法的程序。

36.【★☆☆☆】证明：任何无向连通图至少存在一棵支撑树。

37.【★★★☆】证明：若所有边的权均不相同，则连通带权图中有唯一的最短树。

38.【★☆☆☆】图 3.34 所示的赋权图 G 代表 7 个城市及城市间连接通信的预估造价，给出一个设计方案使得各城市间能够通信且总造价最小，并计算出最小造价。

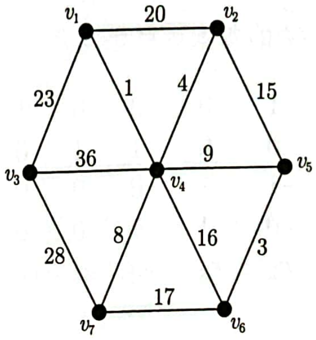

图 3.34

39.【★☆☆☆】求图 3.35 所示带权图中的最短树的边权之和。

40.【★☆☆☆】求权矩阵所示带权图中 (见图 3.36) 的最短树的边权之和。

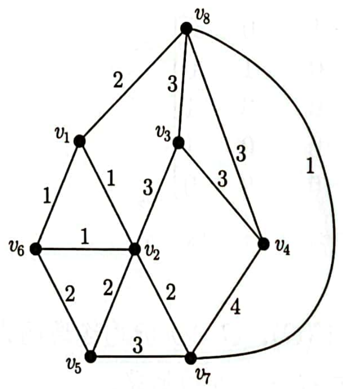

图 3.35

$$\begin{bmatrix} 0 & 3 & 2 & 1 & 3 & 1 & 1 \\ 3 & 0 & 2 & 3 & 4 & 2 & 2 \\ 2 & 2 & 0 & 2 & 2 & 0 & 3 \\ 1 & 3 & 2 & 0 & 1 & 3 & 4 \\ 3 & 4 & 2 & 1 & 0 & 2 & 3 \\ 1 & 2 & 0 & 3 & 2 & 0 & 3 \\ 1 & 2 & 3 & 4 & 3 & 3 & 0 \end{bmatrix}$$

图 3.36

41.【★★☆☆】编写求最短树的程序。

42.【★☆☆☆】求权矩阵所示带权图中的包含边 (v_2, v_5) 最短树的边权之和。

43.【★☆☆☆】求图 3.37 的最短树。

44.【★★☆☆】求图 3.37 所示带权图中的次短树的边权和。

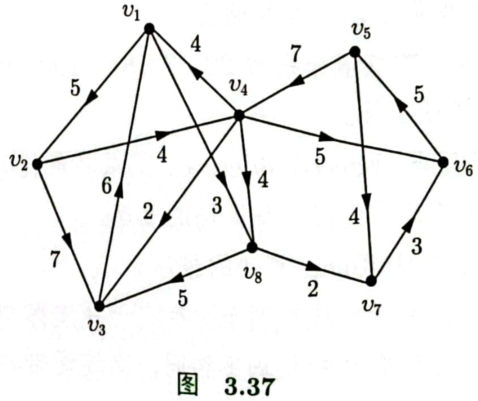

图 3.37