离散数学复习纲要III（图论部分）

第1章. 图论的基本概念(Basic Concepts of Graph Theory)

1. **无向图的基本概念(Basics Concepts for Undirected Graph)**
   (a) **无向图**：称二元组$G=(V,E)$是一个无向图(undirected graph)如果
      i. V是一个非空有限集合，
      ii. E是V中元素的无序对所组成的集合。
      并把V的元素称为图的**顶点**(vertex)，E的元素称为图的**边**(edge)。
   (b) 我们用$V(G)$表示图G的顶点集，用$E(G)$表示图G的边集。若$|V(G)|=n$，则称G为n**阶图**。
   (c) i. **有限图**：V和E是有限集合。
      ii. **无限图**：V和E是无限集合。
   (d) 有限图中：
      $V=\{v_1,v_2,\dots,v_n\},E=\{e_1,e_2,\dots,e_m\}$
      e_k可记为无序或有序点对(v_i,v_j)
      称e_k与v_i,v_j**关联**；e_k**连接**v_i和v_j；v_i,v_j是e_k的**端点**；称v_i和v_j**相邻**。
      我们以后不加说明时，都假定图有n个顶点，m条边。
   (e) i. **孤立点**不与任何边关联的顶点。
      ii. **环**(loop)：两端点重合的边。即$e_k=(v_i,v_i)$。
      iii. **重边**(multiple edges)：若两结点间有两条或两条以上的边，则这些边称为重边。
      iv. **多重图**(multigraph)：有重边的图。
      v. **简单图**(simple graph)：无重边无自环的无向图。
      vi. **空图**(null/empty graph)：无边的简单图，记作N_n。
      vii. **完全图**(complete graph)：任意两结点间都有边的简单图，n个顶点的完全图记作K_n。
      viii. **平凡图**：只有一个顶点的图。
   (f) **度**(degree)：$G=(V,E)$的某结点v所关联的变数称为结点的度，用$d(v)$表示。如果v带有自环，则自环对$d(v)$的贡献为2。
      度为1的顶点称为**悬点**，与悬点关联的边称为**悬边**。度数为奇数的点**奇点**，度为偶数的点称为**偶点**。每个顶点的度都相同的图称为**正则图**,若其顶点的度均为k，则称为k**-正则图**。
   (g) **握手定理**：在无向图$G=(V,E)$中，顶点度数的总和等于边数的两倍，即
      $\sum_{v\in V}d(v)=2|E|$。
      推论：无向图中奇点的数目是偶数。
      推论：非空的简单图中一定存在度相同的点。
      推论：K_n的边树是$\frac{1}{2}n(n-1)$。
2. **图的基本性质(Basic Properties of Graph)**
   (a) **二部图（偶图）**：设G是简单图，若其顶点集V可以划分为两个不相交的非空集合V_1和V_2，而G中所有边都连接V_1中的一个顶点和V_2中的一个顶点（也就是说G中没有边连接着V_1中的两个顶点或V_2中的两个顶点），那么我们将G称作**二部图**或**偶图**。
      若V_1中任一顶点和V_2中每个顶点都相邻，则称二部图G为**完全二部图**或**全偶图**。若$|V_1|=m,|V_2|=n$，则记完全二部图为$K_{m,n}$。
   (b) **子图**：对于图G和图H，
      i. 若$V(H)\subseteq V(G),E(H)\subseteq E(G)$，则称图$H$是图$G$的**子图**(subgraph)，记作$H\subseteq G$。
      ii. 若$H\subseteq G,H\neq G$,则称H是G的**真子图**，记作$H\subset G$。
      iii. 若$H\subseteq G,V(H)=V(G)$，则称H是G的**生成子图**或**支撑子图**(spanning subgraph)。
      iv. 如果$V(H)\subseteq V(G)$，且$E(H)$中包含了G在顶点集$V(H)$中的所有边，则称H是G的

导出子图(induced subgraph)。

v. 平凡子图：G和N_n。

(c) **图的运算**：设$G_1=(V_1,E_1)$和$G_2=(V_2,E_2)$是两个简单图，则：

i. 它们的**并**定义为$G=(V,E)$，其中$V=V_1\cup V_2, E=E_1\cup E_2$，记作$G=G_1\cup G_2$

ii. 它们的**交**定义为$G=(V,E)$，其中$V=V_1\cap V_2, E=E_1\cap E_2$，记作$G=G_1\cap G_2$

iii. 它们的**对称差**定义为$G=(V,E)$，其中$V=V_1\cup V_2, E=E_1\oplus E_2$，记作$G=G_1\oplus G_2$
若G_2是G_1的子图，则定义：

iv. **差**：$G_1-G_2=(V_1,E_1-E_2)$

v. n个结点的简单图G的**补图**$\overline{G}:K_n-G$

vi. 从G中删去结点v及其关联的边：$G-v$

vii. 从G中删去边$e:G-e$

viii. 向G中增加边$e_{ij}=(v_i,v_j):G+e_{ij}$

(d) **图的同构**：对于简单图$G_1(V_1,E_1)$，$G_2(V_2,E_2)$，如果能建立V_1到V_2的双射f，其中G_1中的顶点a和b相邻，当且仅当G_2中的顶点$f(a)$和$f(b)$也相邻，则称G_1与G_2**同构**(isomorphic)，记作$G_1\cong G_2$。
若$G_1\cong G_2$，则有

i. $|V(G_1)|=|V(G_2),|E(G_1)=E(G_2)|$；

ii. G_1和G_2结点度的非增序列相同；

iii. G_1的任一导出子图在G_2中都有与之同构的导出子图；反之亦然。

(e) **赋权图**：如果给图$G=(V,E)$的每条边e_k都赋以一个实数w_k作为该边的**权**(weight)，则称G是**赋权图**。
特别地，如果权都是正数，称为**正权图**。

3. **有向图的基本概念(Basics Concepts for Directed Graph)**

(a) **有向图**：称$D=(V,E)$是一个有向图，如果

i. V是一个非空有限集合；

ii. E是V中元素的有序对所组成的有限集合。并把V的元素叫做图G的**顶点**，E的元素叫做图G的**有向边**或**边**。

(b) i. 设G是有向图，$e=(u,v)$是E中元素，则称u为e的**起点**或**尾**，e为u的**出边**；并称v为e的**终点**或**头**，e为v的**入边**。对于环而言，起点和终点是重合的。

ii. 若两条或两条以上的边有相同的头和尾，则这些边称为**重边**。没有重边和环的有向图称为**简单有向图**。

iii. 若对任意的$u,v\in V$，均有$(u,v)\in E$和$(v,u)\in E$，则称D为**有向完全图**。

iv. 设D为有向简单图，若对任意的$u,v\in V$，有向边$(u,v)\in E$和$(v,u)\in E0$有且仅有一个成立，则称D为**竞赛图**(在清华书中为**完全有向图**)。

(c) 在有向图中，顶点v的出边数称为v的**出度**，记作$d^+(v)$。顶点v的入边数称为v的**入度**，记作$d^-(v)$ 。顶点v的关联边数称作v的**度**，记为$d(v)$，即

$$d(v)=d^+(v)+d^-(v)$$

(d) **定理**：对有向图，有

$$\sum_{v\in V}d^+(v)=\sum_{v\in V}d^-(v)=|E|$$

(e) i. **基础图**: 设D是有向图，若略去边的方向，则得到一个无向图，称其为图D的**基础图**。

ii. **定向图**：如果D是无向图，给所有的边任意定方向后，则得到一个有向图，称为图D的**定向图**。

(f) **邻点集**：

i. 无向图G中,顶点v的邻点集定义为

$$\Gamma(v)=\{u|(v,u)\in E(G)\}$$

      ii. 有向图G中,顶点v的**直接后继集**或**外邻集**定义为

$$\Gamma^+(v) = \{u|(v,u) \in E(G)\}$$

      iii. v的**直接前趋集**或**内邻集**定义为

$$\Gamma^-(v) = \{u|(v,u) \in E(G)\}$$

4. **图的代数表示(Algebraic Representation of Graphs)**
   (a) **邻接矩阵(adjacency matrix)**:
      i. **无向图的邻接矩阵**：设图$G = (V,E)$是一个简单图，$V\{v_1, v_2, \ldots, v_n\}$，$G$的邻接矩阵定义为$\mathbf{A} = (a_{ij})_{(n\times n)}$，其中

$$a_{ij} = \begin{cases} 1, & (v_i, v_j) \in E, \\ 0, & (v_i, v_j) \notin E. \end{cases}$$

      无向图的邻接矩阵总是对称的，邻接矩阵也能用来表示环，但不能表示重边。
      ii. **有向图的邻接矩阵**：设图$D = (V,E)$是一个简单有向图，$V\{v_1, v_2, \ldots, v_n\}$，$D$的邻接矩阵定义为$\mathbf{A} = (a_{ij})_{(n\times n)}$，其中

$$a_{ij} = \begin{cases} 1, & (v_i, v_j) \in E, \\ 0, & (v_i, v_j) \notin E. \end{cases}$$

   (b) **权矩阵**：赋权图常用权矩阵表示:

$$a_{ij} = \begin{cases} w_{ij}, & (v_i, v_j) \in E, \\ 0, & (v_i, v_j) \notin E. \end{cases}$$

   (c) **关联矩阵(incidence matrix)**：
      i. **无向图的关联矩阵**：设图$G = (V,E)$是一个简单无向图，$V\{v_1, v_2, \ldots, v_n\}$，$E = \{e_1, e_2, \ldots, e_m\}$。那么其关联矩阵定义为$\mathbf{M}F = (mG_{ij})_{(n\times m)}$,其中

$$m_{ij} = \begin{cases} 1, & \text{边}e_j\text{和顶点}v_i\text{关联}, \\ 0, & >\text{边}e_j\text{和顶点}v_i\text{不关联}. \end{cases}$$

      关联矩阵也能用来表示有重边的图，但不能表示自环。图中的重边在矩阵中以相同的列矢量表示。
      ii. **有向图的关联矩阵**：对于无环有向图$D = (V,E)$，$V\{v_1, v_2, \ldots, v_n\}$，$E = \{e_1, e_2, \ldots, e_m\}$。那么其关联矩阵定义为$\mathbf{M} = (m_{ij})_{(n\times m)}$,其中

$$m_{ij} = \begin{cases} 0, & \text{边}e_j\text{和顶点}v_i\text{不关联}, \\ 1, & \text{边}e_j\text{是顶点}v_i\text{的出边}, \\ -1, & \text{边}e_j\text{是顶点}v_i\text{的入边}. \end{cases}$$

      iii. **关联矩阵的性质**:
         A. $\sum_{i=1}^{n} m_{ij} = 0, j = 1, 2, \ldots, m$，从而$\sum_{j=1}^{m}\sum_{i=1}^{n} m_{ij} = 0$，说明$\mathbf{M}(D)$中所有元素之和为0。
         B. $\mathbf{M}(D)$中负1的个数等于正1的个数，都等于边数m。
         C. 第i行中，正1的个数等于$d_+(v_i)$，负1的个数等于$d_-(v_i)$。
         D. 每列中只有两个非零元：1和-1。
         E. 关联矩阵能表示重边,但不能表示自环。
         F. 无向图的关联矩阵性质类似，只是没有-1。

第2章. 道路与回路(Path and Circuit)

1. **道路与回路的基本概念(Basic Concepts for Path and Circuit)**
   (a) **有向图中道路与回路：**
      i. **定义**：有向图G中边序列$P=(e_{i_1},e_{i_2},\ldots,e_{i_q})$，其中$e_{i_j}=(v_k,v_l)$满足：$v_k$是$e_{i_{j+1}}$的终点，$v_l$是$e_{i_{j+1}}$的始点，就称$P$是$G$的一条**有向道路**。如果$e_{i_q}$的终点也是$e_{i_1}$的始点，则称$P$是$G$的一条**有向回路**。
      ii. **简单有向道路/回路**：P中边不重复出现。
      iii. **初级有向道路/回路**：P中结点不重复出现。
   (b) **无向图中道路与回路：**
      i. **定义**：无向图G中，若点边交替序列$P=(v_{i_1},e_{i_1},v_{i_2},e_{i_2},\ldots,e_{i_{q-1}},v_{i_q})$，满足：$v_{i_j}$和$v_{i_{j+1}}$是$e_{i_j}$的两个端点，则称$P$是$G$中的一条**链**或**道路**。如果$v_{i_q}=v_{i_1}$，则称$P$是$G$中的一个**圈**或**回路**。
      ii. **简单道路/回路**：P中没有重复出现的边。
      iii. **初级道路/回路**：P中没有重复出现的结点。
      iv. **注**：道路又称**通道**；回路又称**闭通道**（开通道即非回路的道路）；简单道路又称**迹**；初级道路又称**路**、**路径**；初级回路有时也称为**圈**。
   (c) **周长和围长：**
      i. 长度为奇数的圈（初级回路）称为**奇圈**；长度为偶数的圈（初级回路）成为**偶圈**。
      ii. 在含圈的无向简单图G中，称G中最长圈的长度为G的周长，记做$c(G)$，G中最短圈的长度为G的围长，记做$g(G)$。
      iii. 无向完全图$K_n(n\geq 3)$的周长为n，围长为3。完全二部图$K_{n,n}(n\geq 2)$的周长为$2n$，围长为4。
   (d) **简单图中的弦：**
      i. **定义**：设C是简单图G中含结点数大于3的一个初级回路。如果结点v_i和v_j在C中不相邻，而边$(v_i,v_j)\in E(G)$，则称(v_i,v_j)是C的一条**弦**。
      ii. **定理**：若对每一个$v_k\in V(G)$，都有$d(v_k)\geq 3$，则G中必含带弦的回路。
   (e) **二分图的回路**：如果二分图G中存在回路，那么它们都是由偶数条边组成的。
2. **图的连通性(connectivity)**
   (a) **连通性定义：**
      i. 若无向图G的任意两个结点之间都存在道路，就称G是**连通的(connected)**。
      ii. 对有向图G，若不考虑边的方向时是连通的，则称G是**(弱)连通的**。
   (b) **连通分支(connected component)：**
      i. 若G的连通子图H不是G的任何连通子图的真子图，则称H是G的极大连通子图，或称**连通分支**，又称**连通支**、**连通分量**。
      ii. 显然G的每个连通分支都是它的导出子图。

iii. 任何非连通图都是2个以上连通支的并。

(c) 如果连通图中不含回路，而且任意两个结点间都只有唯一的一条初级道路，则称该连通图为**树**。树是边数最少的连通图。

(d) **距离**：设u, v为图G中的任意两个顶点，若u, v连通，称设u, v之间长度最短的通道为u, v之间的**短程线**，短程线的长度称为u, v之间的距离，记做$d(u, v)$，当u, v不连通时，定义$d(u, v) = \infty$。

(e) **定理**：设$G = (V, E)(|V| = n, |E| = m)$是连通图，$m \geq n - 1$。

(f) **路与圈的相关定理**：

i. **定理**：设$G = (V, E)(|V| = n)$,若顶点u和$v(u \neq v)$之间有通道，则它们之间存在长度小于n的通道。

ii. **推论**：图$G = (V, E)(|V| = n)$中，若顶点u和$v(u \neq v)$之间有通道，则它们之间存在长度小于n的路。

iii. **定理**：图$G = (V, E)(|V| = n)$中，若存在顶点u到自身的闭通道，则一定存在u到自身的长度小于等于n的闭通道。

iv. **推论**：图$G = (V, E)(|V| = n)$中，若存在顶点u到自身的闭迹，则一定存在u到自身的长度小于等于n的闭迹。

v. **扩大路法**：图论中很有用的证明方法：设为$G = (V, E)$为n阶无向图，$E \neq \emptyset$，设Γ_k为G中一条路，若此路的始点或终点与通道外的顶点相邻，就将它们扩到通道中来，继续这一过程，直到最后得到的通道的两个端点不与通道外的顶点相邻为止，设最后得到的路径为Γ_{k+r}（长度为k的路扩大成了长度为$k + r$的路径），称Γ_{k+r}为“**极大路**”，称使用此种方法证明问题的方法为“**扩大路法**”。

vi. **定理**：一个图为二部图当且仅当图G中无奇圈。

(g) **割点、割边与桥**：如果移除某个顶点及所有与该顶点相关联的边所产生的子图比原图有更多的连通分量，这样的顶点称为**割点**。显然，从一个连通图中移除一个割点就会产生一个不连通的子图。类似地，将一条边移除后所产生的子图比原图有更多的连通分量，这样的边称为**割边**或**桥**。

3. 欧拉道路与欧拉回路(**Euler Path and Euler Circuit**)

(a) **定义**：给定无向连通图$G = (V, E)$，包含图G的所有边的简单道路称为**欧拉道路**(或**欧拉通道、欧拉迹**)，包含图G的所有边的简单回路称为**欧拉回路**(或**欧拉闭迹**) 。假设G没有孤立点，若G含有欧拉回路，则称G是**欧拉图**。

(b) **定理**：图G是欧拉图的充要条件是G连通且没有奇点。

(c) **推论**：若有向连通图G中各结点的正负度相等，则G中存在有向欧拉回路。

(d) **推论**：若无向连通G中只有2个度为奇数的顶点，则G中存在欧拉道路。

(e) **一笔画问题**：如果一个图可以从某个顶点出发，每条边恰好经过一次，最后终止在出发点或另一个顶点，则称该图可以**一笔画**。也就是说，可以一笔画的图含有欧拉道路，但不一定有欧拉回路。显然欧拉图一定可以一笔画成，反之则一般不然。

(f) **定理**：一个图能一笔画成的充要条件是，G连通且奇顶点数为0或2。若奇点数为2，则从其中一个奇顶点出发，终止于另一个奇顶点上，完成一笔画。

4. 哈密顿道路与哈密顿回路(**Hamilton Path and Hamilton Circuit**)

(a) **定义**：无向图G的一条经过全部结点的初级回路(道路)称为G的**哈密顿回路(道路)**。简

记为H回路(道路)。含有哈密顿圈的图称为**哈密顿图**，反之则称为**非哈密顿图**。

(b) **H回路判定：**

i. **定理：**若简单图G的任意两结点v_i与v_j之间恒有$d(v_i)+d(v_j)\geq n_1$，则G中存在H道路。

ii. **推论**(Ore 定理)：若简单图$G(n\geq 3)$的任一对不相邻结点v_i与v_j都满足$d(v_i)+d(v_j)\geq n$则G有H回路。

iii. **推论**(Dirac 定理)：若简单图$G(n\geq 3)$的任一结点的度$\geq \frac{n}{2}$，则G有H回路。

iv. **引理：**设G是简单图，v_i,v_j是不相邻顶点，且满足$d(v_i)+d(v_j)\geq n$，则G存在H回路的充要条件是$G+(v_i,v_j)$有H回路。

(c) **闭合图：**

i. 若v_i和v_j是简单图G不相邻的结点，且满足$d(v_i)+d(v_j)\geq n$，则令$G'=G+(v_i,v_j)$。对G'不断加入这样的边(v_i,v_j),直至不再有满足条件的结点对,最终得到的图称为G的闭合图,记作$C(G)$。

ii. **引理：**简单图G的闭合图$C(G)$是唯一的。

iii. **定理**(Bondy&Chvátal,1976)：简单图G有H回路当且仅当$C(G)$有H回路。

iv. **推论：**设$G(n\geq 3)$是简单图，若$C(G)=K_n$，则G有H回路。

(d) **连通分量定理：**

i. **定理：**设G是哈密顿图，则对于G顶点集的任意子集S，在G中移除S中的顶点及所有与这些顶点相关联的边，所产生子图的连通分量数必不大于$|S|$。即$\forall S\subseteq V, p(G-S)\leq |S|$，其中$p(G-S)$为$G-S$的连通分支数。

ii. **推论：**每个哈密尔顿图都没有割点。

iii. **推论：**有奇数个顶点的二部图必定不是哈密顿图。

第3章. 树(Tree)

1. **树的基本概念(Basic Concepts of Trees)**

(a) **基本定义：**

i. 无回路的连通无向图称为**无向树**，简称**树**。常用T表示

ii. 树中最长路的长称为树的**高**。

iii. 树的度为1的顶点称为**树叶**，其余的顶点称为**分枝点**。树的边称为**树枝**。

iv. 平凡图称为**平凡树**。

v. 若无向图G至少有两个连通分支，每个连通都是树，则称G为**森林**，简称**林**。

vi. 由定义，森林即每个连通分量都是树的图。树的树叶就是**悬点**。

(b) **割边：**

i. **定义：**设e是G的一条边，若$G'=G-e$比G的连通分支数增加，则称e是G的一条**割边**。

ii. 树的每条边都不属于任何回路，都是割边。

iii. **定理：**$e=(u,v)$是割边，当且仅当e不属于G的任何回路。

(c) **树的基本性质：**

i. **定理：**设$G=(V,E)$非平凡无向图，边数为m，顶点数为n，则下列命题等价：

A. G是树；

B. G连通且无回路；

C. G中任意两个顶点之间存在唯一的路径。

D. G无圈且$m=n-1$；

E. G连通且$m=n-1$；

F. G是连通的且G中任何边均为割边。
G. G中没有圈，但在任何两个不同的顶点之间加一条新边，在所得图中得到唯一的一个含新边的圈。

ii. **推论**：非平凡树至少有两片树叶。
iii. **推论**：树T中一定存在树叶结点。

2. **有根树(Tree with Fixed Root)**
   (a) **有根树的基本概念**：
      i. **有根树的构造**：
         A. 在很多应用中，常指定树中某特定顶点作为树根。
         B. 指定根后，设树的根为r，所有与r相邻的顶点构成集合V_0，对于所有$v \in V_0$，为边(r, v)指定方向，将r作为起点，v作为终点。
         C. 然后找到除r以外所有与V_0中元素相邻的顶点，构成顶点集V_1，为所有连接V_0中元素和V_1中元素的边指定方向，将V_0中元素作为起点，V_1中元素作为终点，继而找到所有未访问过的与V_1中元素相邻的顶点，构成V_2。
         D. 依次类推，直到不存在未访问过的与V_k相邻的顶点时过程结束。
      ii. **定义**：根据以上过程产生的有向图称为**有根树**。我们可以任意指定一个顶点为根来将一个无根树转化为有根树，而选择不同的根将产生不同的有根树。
      iii. **有根树中顶点的关系**：
         A. 在有根树T中，设v是非根顶点，对于有向边(u, v)，将顶点u称为v的**父顶点**。显然u是唯一的，否则产生圈。相对地，将v称为u的**子顶点**。
         B. 有相同父顶点的顶点称为**兄弟顶点**。若存在从u到v的有向路，称u为v的**祖先顶点**，称v为u的**后代顶点**。没有子顶点的顶点称为**树叶**。有子顶点的顶点称为**内顶点**。
      iv. 若v是有根树T中的一个顶点，那么由v和它的所有后代及所有与这些后代相关联的边所形成的子图称为以v为根的**子树**。
      v. 从根到顶点v的有向路的长度称为v的**层数**。树叶顶点的最大层数称为有根树的**高**，换言之，有根树的高就是就是从根到其他所有顶点的有向路的最大长度。
   (b) **有序树**：
      i. **定义**：设T是一棵有根树，若将T中层数相同的顶点都标定次序，则称T为**有序树**。
      ii. **分类**：根据根树T中每个分支点儿子数以及是否有序，可以将根树分成下列各类：
         A. 若T的每个分支点至多有m个儿子，则称T为m**元树**；又若m元树是有序的，则称它为m**元有序树**。
         B. 若T的每个分支点都恰好有m个儿子，则称T为m**元正则树**；又若T是有序的，则称它为m**元正则有序树**。
         C. 若T是m元正则树，且每个树叶的层数均为树高，则称T为m**元完全正则树**，又若T是有序的，则称它为m**元完全正则有序树**。
   (c) **根树的遍历**：对一棵有根树的每个顶点都访问且仅访问一次称为**遍历一个根树**。若一棵高度为h的有根m元树的所有树叶都在h层或$h-1$层，则这棵树是**平衡的**。
   (d) **有根树的相关定理**：
      i. **有根树结点定理**：带有i个内顶点的m元正则树含有$n = mi + 1$个顶点。
      ii. m**元正则树结点定理**：一个m元正则树，满足
         A. n个顶点有$i = \frac{n-1}{m}$个内顶点和$l = \frac{(m-1)n+1}{m}$个树叶。
         B. i个内顶点有$n = mi + 1$个顶点和$l = (m-1)i + 1$个树叶。
         C. l个树叶有$n = \frac{ml-1}{m-1}$个顶点和$i = \frac{l-1}{m-1}$个内顶点。
      iii. m**元树定理**：在高度为h的m元树至多有m^h个树叶。

3. **二叉树和和哈夫曼树(Binary Tree and Huffman Tree)**
   (a) **二叉树**：

i. **定义**：二叉树是二元有序树，即每个顶点的出度不大于2，且每个顶点的儿子按一定顺序排列的有根树。
ii. **完全二叉树**：每个顶点的出度为0或2的二叉树称为**二叉正则树**或**完全二叉树**。

(b) **二叉树定理**：

i. 在二叉树中，第i层顶点总数不超过2^{i-1}。
ii. 深度为h的二叉树最多有$2^h-1(h\geq 1)$个顶点。
iii. 对于任意一棵二叉树T，如果叶顶点数为N_0，而出度数为2的顶点总数为N_2，则$N_0=N_2+1$。

(c) **二进制编码**：

i. **问题**：在计算机和通讯中常用二进制编码表示字符，我们可以利用二叉树来对字符进行二进制编码。构造一棵二元树，用其树叶来表示所要编码的字符。从根出发，经过一条有向路到达某一树叶，在该有向路上，规定从一个内顶点左下行表示0，右下行表示1，于是整条有向路就给出了一个二进制串。而这个二进制串就是该树叶所表示的字符的二进制编码。
ii. **前缀码**：设$\alpha_1\alpha_2\ldots\alpha_{n-1}\alpha_n$是长为$n$的符号串，称其子串

$$\alpha_1,\alpha_1\alpha_2,\alpha_1\alpha_2\ldots\alpha_{n-1}$$

分别是它的长为$1,2,\ldots,n-1$的**前缀**。设$A=\{\beta_1,\beta_2,\ldots,\beta_m\}$是$m$个符号串的集合，若对任意的$\beta_i,\beta_j\in A,(i\neq j)$，它们互不为前缀，则称$A$为**前缀码**。若$A$中的符号串只出现0和1两个符号，则称$A$为**二元前缀码**。
iii. **前缀码定理**：一棵二叉树产生一个二元前缀码，并且由二元正则树产生的二元前缀码是唯一的。
iv. **编码的最佳标准**：使得在编码$A=\{\beta_1,\beta_2,\ldots,\beta_m\}$中，设符号串$\beta_i$的出现的几率为$p_i$，长度为$l_i$，则$L=\sum_{i-1}^{m}p_il_i$达到最小。

(d) **哈夫曼树**：

i. **定义**：设T为二叉树，若t片树叶v_i分别对应实数w_i，则称w_i是树叶v_i的**权**，其中$i=1,2,\ldots,t$，而$w(T)=\sum_{i=1}^{t}w_il_i$称为$T$的权，其中$l_i$是树叶$v_i$的层数，并称$T$为**带权二叉树**。在所有带权$w_1,w_2,\ldots,w_t$的$t$片树叶的二叉树中，权最小的二叉树称为**最优二叉树**，又叫**Huffman树**。
ii. **Huffman算法**：
(1) 根据给定的n个权值$\{w_1,w_2,\ldots,w_n\}$构成n棵二元树的集合$F=\{T_1,T_2,\ldots,T_n\}$，其中每棵二叉树T_i中只有一个带权w_i的根，其左右子树都为空。
(2) 在F中选取两棵根的权值最小的树作为左右子树构造一棵新的二叉树，且置新二叉树的根的权值为其左右子树上根的权值之和。
(3) 在F中删除这两棵树，同时将新得到的二叉树加入F中。
(4) 重复(2)和(3)，直到F只含一棵树为止。这棵树就是Huffman树。
iii. **定理**：由Huffman算法得到的二叉树是最优二叉树。

4. **生成树(Spanning Tree)**

(a) **生成树的基本概念**：

i. **定义**：设T是无向图G的子图并且为树，则称T为G的**树**。若T是G的树且为生成子图，则称T是G的**生成树**(**支撑树**)。设T是G的生成树。$\forall e\in E(G)$，若$e\in E(T)$，则称e为T的**树枝**，否则称e为T的**弦**。称导出子图$G[E(G)-E(T)]$为T的**余树**，记作$\overline{T}$。一般情况下，余树不是一棵树。
ii. **定理**：无向图G连通当且仅当G有生成树。
iii. **推论**：设G为n阶m条边的无向连通图，则$m\geq n-1$。
iv. **推论**：设G是n阶m条边的无向连通图，T为G的生成树，则T的余树中含有$m-n+1$条边（即T有$m-n+1$条弦）。
v. **推论**：设T是连通图G的一棵生成树，$\overline{T}$为T的余树，C为G中任意一个圈，则$E(\overline{T})\cap E(C)\neq\emptyset$。

(b) **图的遍历：**

i. **广度优先搜索算法：**简记为**BFS(Breadth-First-Search)**

(1) 算法求解时常假设给定图G的初态是所有顶点均未曾访问过。在G中任选一顶点s为初始点(源点)。从顶点s开始标记，将s标记为0。

(2) 找出与s相邻的顶点，给它们做标记$1(s)$

(3) 考虑每个与标记为1的顶点v相邻的未标记的顶点，把这些顶点加1，标记为$2(v)$。

(4) 按(3)的方式继续进行，直到G中没有与已标记顶点相邻的未标记顶点为止，算法结束。

(c) **最小生成树(MST)：**设无向连通带权图$G =< V, E, W >$，T是G的一棵**生成树**。T的各边权之和称为T的权，记作$W(T)$。G的所有生成树中权最小的生成树称为G的**最小生成树(Minimum Spanning Tree, MST)**或(**最短树**)。

(d) **Kruskal算法(避圈法)：**

i. **算法流程：**设n阶无向连通带权图$G =< V, E, W >$有m条边。

(1) 将m条边按权从小到大顺序排列，设为$e_1, e_2, \ldots, e_m$，令$T = \emptyset$。

(2) 取e_1在T中，然后依次检查$e_2, e_3, \ldots, e_m$，若e_j与T中的边不能构成圈，则取e_j在T中，否则弃去e_j。

(3) 依次加入$n-1$条边，算法停止时得到的T为G的最小生成树。

ii. **定理：**(正确性)$T = (V, E')$是赋权连通图$G = (V, E)$的最小生成树，当且仅当对T的任意余树边$e \in E - E'$，初级回路$C^e(C^e \subseteq E' + e)$满足其边权

$$w(e) \geq w(a), a \in C^e(a \neq e)$$

。

iii. **定理：**Kruskal算法的结果一定得到赋权连通图G的一棵最小生成树。

iv. **定理：**(复杂性)Kruskal算法的计算复杂性是$O(m + p \log m)$，其中p是迭代次数。

v. 适合于边稀疏的情况。

(e) **Prim算法：**

i. **算法流程：**设$G =< V, E, W >$是有n个顶点的连通无向图。

(1) 初始化集合$U = \emptyset$和$T = \emptyset$，从V中随机选择一个顶点v加入到集合U中。

(2) 选择与U中顶点相邻权值最小的边的另一顶点v，把v加入到U中，并将该边加入T中。

(3) 重复执行(2),直到$U = V$为止。此时的T为G的最小生成树。

ii. **定理：**(正确性) 设V'是赋权连通图$G = (V, E)$的结点真子集，e是二端点分跨在V'和$V - V'$的权值最小边，则G中一定存在包含e的最小生成树T。

iii. **定理：**Prim算法的结果一定得到赋权连通图G的一棵最小生成树。

iv. 适合于边稠密的情况。