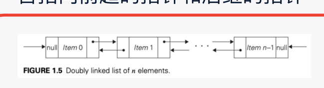
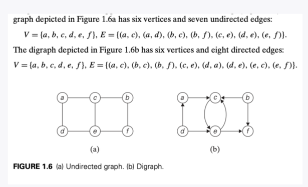
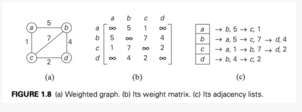
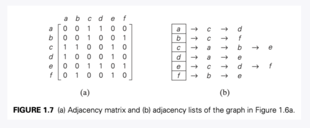
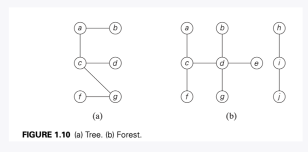
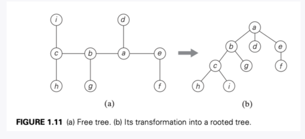
基本数据结构

- **线性数据结构 Linear Data Structures**
  - 数组Array
    - n个相同数据类型的元素构成的序列，它们连续存储在计算机的存储器中
    - i.e. 字符串
    - 
    - example
  - 链表Linked List
    - 0个或多个节点node的元素构成的序列
      - 1. 数据data
      - 2. 一个或多个指针pointer
    - 种类
      - 1. 单链表 singly linked list
        - 特点：特殊节点 表头header，包含链表的信息，如链表的长度，指向头元素的指针，指向尾元素的指针
        -
        - example
      - 2. 双链表 doubly linked list
        - 特点1. 除了首尾两个节点，每一个节点都同时包含指向前趋的指针和后继的指针
        - 
        - example
  - 栈Stack
    - 操作在端部进行，这一端称为栈顶top：进栈（压入）push onto，出栈（弹出）pop off
    - LIFO
  - 队列Queue
    - 操作在列表的两头进行：出队dequeue，这一头称为队头front；入队enqueue，这一头称为队尾rear
    - FIFO
    - 优先队列priority queue：动态改变的候选集合中选择一个优先级最高的元素。实现：堆heap
- **图 Graph**
  - 定义：图 G = <V, E>
    - 有限集合V，它的元素称为 顶点 vertex
    - 有限集合E，它的元素是一对顶点，称为 边 edge
    - 只考虑不含圈的图； 圈loop：连接顶点自身的边
    - 
    - example
  - 种类
    - 无向图 undirected graph
      - 每对顶点没有顺序，(u,v) = (v,u)，顶点u和顶点v相互邻接 adjacent，它们通过无向边 undirected edge (u,v)相连接
      - 顶点u和顶点v 称为边(u,v)的 端点 endpoint
      - 顶点u和顶点v与边(u,v)相 关联 incident，vice versa
      - 定义不允许无向图的同一对顶点拥有多条边，对于|V|个顶点的无圈无向图，它可能包含的边数|E|
        - 0 <= |E| <= |V| (|V| - 1) / 2
    - 有向图 directed graph
      - 顶点对(u,v)不同于顶点对(v,u)，边(u,v)的方向是从顶点u到顶点v，u称为尾tail，v称为头head
    - 完全图 complete graph
      - 定义：任意两个顶点之间都有边相连的图
      - i.e. |V|个顶点，标准符号K |V|
      - 稠密图 dense：图中所缺的边数量相对较少
      - 稀疏图 sparse：图中边相对顶点来说数量较少
    - 加权图 weighted graph / 加权有向图 weighted digraph
      - 定义：一种给边赋了值的图，这些值称为边的权重 weight 或 成本 cost
      - i.e. 寻找交通网络/通信网络两点间的最短路径
      - 表示
        - 邻接矩阵：不存在这条边：∞；这种矩阵又称为权重矩阵 weighted matrix 或 成本矩阵 cost matrix
        - 邻接链表
        - 
        - example
  - 图的表示
    - 邻接矩阵 adjacency matrix： n x n 的布尔矩阵，有连接边：1；无连接边：0
      - 无向图的邻接矩阵总是对称的
    - 邻接链表 adjacency list：每一个顶点用一个邻接链表表示，该链表包含了这个顶点邻接的所有顶点
    - 
    - example
  - 路径paths 和 环 cycles
    - 定义：从顶点u到顶点v的路径是图G中 始于u之于v的邻接顶点序列
    - 图的两个特性：连通性 connectivity 和 无环性 acyclicity
      - 连通性 connectivity
        - 如果对于图中的每一对顶点u和v 都有一条从u到v的路径，我们说该图是连通的 connected
        - 如果图是非连通的，这样的模型包含 几个自我连通的部分，称为该图的 连通分量 connected component；连通分量是给定图的极大连通子图；i.e. 美国州际公路系统示意图
      - 无环性 acyclicity
        - 回路 cycle：它的起点和终点都是同一顶点，长度大于0，而且绝不会将同一条边包含两次
        - 不包含回路的图称为无环图 acyclic graph
    - 简单路径 simple path：一条路径上所有的顶点都是互不相同的
    - 路径的长度 length：将路径代表的顶点序列中的顶点数目-1，恰好与路径包含的边的数目一致
    - 有向路径 directed path：是顶点的一个序列，序列中的每一对连续顶点都被一条边连接起来，边的方向等于从第一个顶点指向下一个顶点的方向
- **树 Tree**
  - 定义：自由树 free tree 就是连通无回路的图
    - 无回路但不一定连通的图称为 森林 forest，它的每一个连通分量是一颗树
    - i.e. 典型应用：描述层次关系：文件目录/企业的组织架构/ 字典的实现
    - 
    - example
  - 特征
    - 特征 1：树的边数总是比它的顶点数少1
      - |E| = |V| - 1
      - 使图成为树，这个特征是必须的，但不是充分的
      - 对于连通图，这个特征是充分的，还可以作为检验连通图是否包含回路的简便方法
    - 特征 2: 树的任意两个顶点之间总是恰好存在一条从一个顶点到另一个顶点的简单路径。
  - 树中的关系
    - 根 root
      - 任选自由树中的一个顶点，放在最顶上（第0层），邻接的顶点放在根的下面
      - 树的 高度height 和 深度 depth
        - 顶点v的深度是从根到v的简单路径的长度
          - 根的层数0，顶点的深度就是在树中的层数
        - 树的高度是从根到叶节点的最长简单路径的长度
          - 树的高度 就是 顶点的最大层数
    - 祖先 ancestor
      - 对于树T中的任意顶点v，从根到该顶点的简单路径上的所有顶点都称为v的祖先；一般也将顶点本身作为它自己的祖先
      - 真祖先 proper ancestor：顶点本身以外的所有祖先顶点的集合称为真祖先集合
    - 父母 parent
      - 如果 (u,v)是从根到顶点v的简单路径伤的最后一条边，则u称为v的父母
      - 至少有一个子女的顶点称为父节点 parental
    - 子女 child
      - 如果 (u,v)是从根到顶点v的简单路径伤的最后一条边，则v称为u的子女
      - 没有子女的顶点称为叶节点 leaf
    - 兄弟 sibling
      - 具有相同父母的顶点 称为兄弟
    - 子孙 descendant
      - 所有以顶点v为祖先的顶点称为v的子孙
      - 真子孙 proper descendant：顶点v的真子孙不包括顶点v
      - 顶点v的所有子孙以及连接子孙的边构成了T的以v为根的子树 subtree
  - 种类
    - 1. 有根树 rooted tree
      - 任选自由树的一个顶点 将它作为有根树的根
      - 状态空间树 state-space tree：强调两种重要的算法设计技术：回溯和分支界限 （chapter 12)
      - 
      - example
      -
      - relations example
    - 2. 有序树 ordered tree
      - 有序树是一棵有根树，树中每一顶点的所有子女都是有序的
        - 为了方便，假设这种树中，所有的子女都是从左到右有序排列
      - 实例 1: 二叉树 binary tree
        - 二叉树的所有顶点的子女个数都不超过2个，并且每个子女不是父母的左子女 left child 就是 父母的右子女 right child
          - 二叉查找树 binary search tree：每个父母顶点的数字逗比它左子树中的数字大，比右子树的数字小
            - 可被推广为更一般的查找树，称为多路查找树 multiway search tree
        - 二叉树也可能为 空
        - 对于高度为h，具有n个顶点的二叉树
          - [log n] <= h <= n -1
- **集合与字典 Set & Dictionary**
  - 集合
    - 定义：它是互不相同项的无序组合（也可以为空），这些项称为集合的元素 element
    - 一个特定的集合：1. 直接列出元素的确切列表；2. 指出集合的特殊属性
    - 实现
      - 1. 只考虑一些称为通用集合 universal set 的大集合 U 的子集
      - 2. 用线性列表的结构来表示集合元素
        - 与列表的差别
          - 1. 集合不能包含相同的元素， 列表可以
            - 绕过唯一性要求：引进 多重集 multiset 或者 包 bag：多重集和包 是可重复项的无序组合
          - 2. 集合是元素的无序组合， 列表定义为元素的有序组合
  - 字典
    - 对集合/多重集做的最多的操作：从集合中查找一个给定的元素，增加新元素，和 删除一个元素。 能够实现这三种操作的数据结构称为字典。
    - 实现
      - 数组
      - 类似散列法
      - 平衡查找树

XMind