

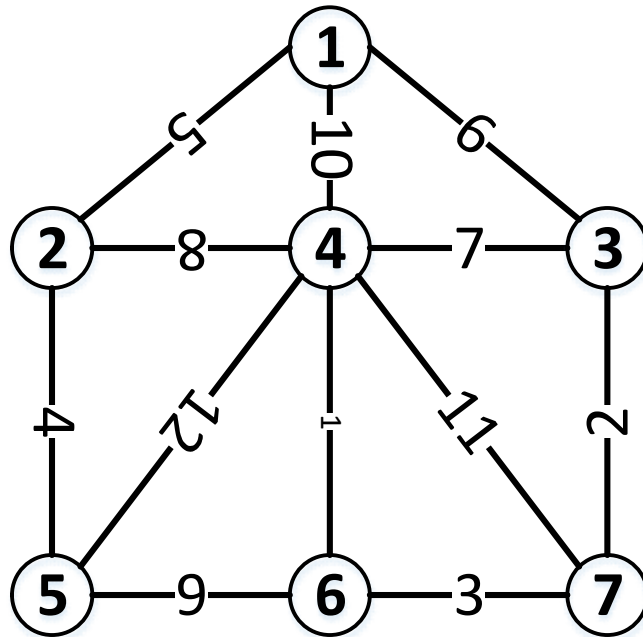
1. 假设结点序列  $F=(60,30,90,50,120,70,40,80)$ , 试用查找树的插入算法, 用  $F$  中的结点依次进行插入, 画出每一步插入后的查找树。再用查找树的删除算法, 从查找树中依次删除 40, 70, 60, 画出删除后的查找树。
2. 试用 Adelson 插入方法依次把结点 50, 20, 10, 100, 120, 30, 110, 60, 70, 90, 80, 40 插入到初始为空的平衡查找树中, 使得每次插入后保持该树仍然是平衡查找树。请一次画出每次插入后形成的平衡查找树。
3. 假设给定结点序列为  
(200, 100, 250, 150, 120, 110, 220, 205, 210, 090, 160, 080, 170, 202, 225, 240, 245)。  
首先在初始时为空的 5 阶 B-树中, 按上面给出的结点序列依次插入, 试画出每次插入后的各棵 5 阶 B-树。然后, 依次删除 202, 150 和 200, 试画出每次删除后的各棵 5 阶 B-树。
4. 设结点  $k_0$ 、 $k_1$ 、 $k_2$ 、 $k_3$ 、 $k_4$  的键值分别为的相对使用频率分别为 5、6、3、7、4。试画出用 Huffman 算法构造出由  $k_0$ 、 $k_1$ 、 $k_2$ 、 $k_3$ 、 $k_4$  组成的 Huffman 树, 并给出这五个键值的编码。
5. 在初始为空的 trie 结构中, 首先依次插入 over, overbalance, overbear 和 overbearing, 然后再依次删除 overbalance, overbearing 和 over。试画出每次插入和删除后的各棵 trie 树。

1. 对于下面的带权无向图，按照

(1) Kruskal 算法

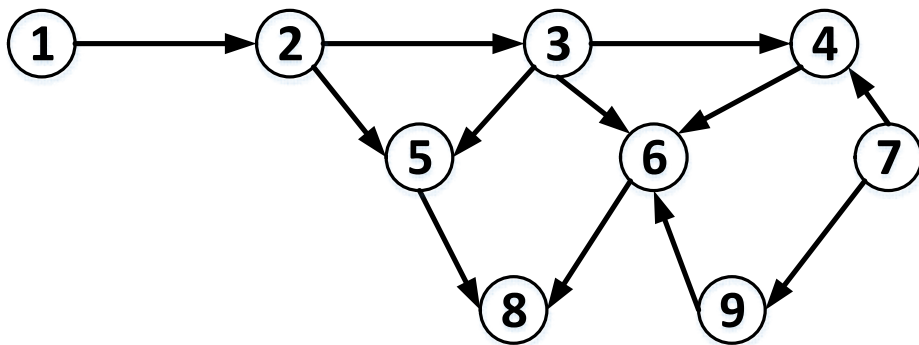
(2) prim 算法（假设以顶点 1 作为出发顶点）

分别给出一棵最小代价生成树，并且用图的序列来表明最小代价生成树的行程过程。



2. 根据 Kruskal 算法，编写一个在连通的带权无向图中寻找最小代价生成树的程序。

3. 在给定的有向图中，用拓扑排序方法求得的顶点的拓扑序列不是唯一的。对下面的有向图的顶点进行拓扑排序，写出任意一种拓扑序列。



4. 下图的 AOE 网络表示一个工程的进度计划，图中边上的权表示活动所需的时间。

- (1) 求出每个顶点所表示的事件的最早发生时间和最迟发生时间。
- (2) 求出每个活动的最早开始时间和最迟开始时间。
- (3) 完成改工程需要多少天？
- (4) 哪些活动是关键活动？
- (5) 给出所有关键路径。

