

题号	一	二	三	四	五	六	七	八	九	十	总分	总分人
得分												

得分	阅卷人

一、线性结构 (20分)

1. 在如下数组中存储了一个线性表, data 表示数据元素, next 表示该元素的后继元素, 表头指针为 A[0].Next, 试写出该线性表中元素的逻辑顺序。

A 0 1 2 3 4 5 6 7

Data 68 55 78 90 2 4

Next 3 5 7 2 0 4 1

2. 一列数据: {10,23, 46, 57,63, 40, 80,25, 18,36}, 对其用散列表存储, 设散列函数为 $H(key) = key \% 13$, 并用线性开型方式解决冲突, 请画出散列表, 并给出查找 18、55 的查找次数。

3. 设一组初始关键字序列为 {20,18, 22, 16,30, 19,28, 8}, 分别用冒泡法排序、插入排序、快速排序 (第一个元素 20 为轴元素), 给出进行一趟的结果。

4. 设一个有序单链表中有 n 个节点, 现要求插入一个节点后仍保持有序, 操作的时间复杂度为: _____。

5. 设一个输入序列为 1、2、3、...、 n , 经过栈的调整, 第一个输出元素为 n , 则第 i ($1 < i < n$) 个输出元素为: _____。

得分	阅卷人

二、层次结构 (25分)

1. 请给出根据二叉树的后序遍历与中序遍历求 二叉树 的算法思想。

2. 给定一个关键字序列 {20, 20, 12, 15, 21, 40, 22}, 请问是否最大堆, 若不是, 请调整成最大堆。

3. 给定一组数列 {7,6,2,1,4,3,9} 分别代表字符 A,B,C,D,E,F,G 出现的频度, 试叙述建立霍夫曼树的算法思想, 画出霍夫曼树, 给出各字符的编码值, 并说明这种编码的优点。

4. 3 阶 B-树依次插入 3,2,1,4,5,6, 9, 再按照同样的顺序删除。依次画出 B-树的变化。

5. 请给出 h 层二叉树的节点数范围。

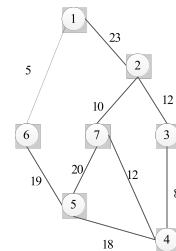
得分	阅卷人

三、网状结构 (25分)

1、(10分)

(1) 给出 prim 算法的思想

(2) 用 prim (以顶点 5 为起点) 算法计算出如下连通网的最小生成树, 给出计算过程。



2 (10分) 下面是某有向加权图(顶点 A,B,C,D,E)的耗费邻接矩阵。

(1) 写出该图的一个拓扑序列,

(2) 使用 Dijkstra 算法依次计算出顶点 A 至其它各顶点的最短路径和最短路径长度。

	A	B	C	D	E
A	0	6	60	20	∞
B	∞	0	∞	9	∞
C	∞	∞	0	∞	5
D	∞	∞	30	0	40
E	∞	∞	∞	∞	0

3. (5分) 对于 n 个顶点的有向图, 用邻接链表表示, 简述如何求解下列问题:

(1) 图中有多少条边?

(2) 顶点 i 的入度和出度是多少?

得分	阅卷人

四、算法题 (30分)

1. 已知一个 LinkList 类型的单链表 L (带头结点), 其中每个结点的类型为 LinkNode。LinkNode 中含有两个域: data 域和 next 域, 其中 data 域为整型元素, next 域为指针域, 指向其直接后继结点。按如下提示的函数原型编写算法, 删除 L 中 data 域取值大于 s, 小于 t 的所有结点, 说明参数值 $s < t$ 。

```
void DeleteElem( LinkList &L, int s, int t)
```

```
{
```

```
}
```

2. 已知有向图 g 采用邻接表的形式存放, 有向图中有 n 个顶点, 判断顶点 V_i 到 V_j 是否有通路 (

$1 \leq i < j \leq n$), 有返回 1, 没有返回 0。其中表头结点包含 data 和 firstarc 分别表示数据域和链域, 表结点包含 adjvex 和 nextarc 分别表示邻接点域和链域。(注意: 本题要求采用非递归算法求解)

```
int count( AdjList g, vertype  $V_i, V_j$ )
```

```
{
```

```
}
```

1. 在如下数组中存储了一个线性表，data 表示数据元素，next 表示该元素的后继元素，表头指针为 A[0].Next，试写出该线性表中元素的逻辑顺序。

A	0	1	2	3	4	5	6	7
Data	68	55	78	90	2			4
Next	3	5	7	2	0	4		1

68 → 2 → 90 → null → 78 → 55 → 4

2. 一组数据：(10,23, 46, 57,63, 40, 80,25, 18,36)，对其用散列表存储，设散列函数为 $H(key) = key \% 13$ ，并用线性开型方式解决冲突，请画出散列表，并给出查找 18、55 的查找次数。

0	1	2	3	4	5	6	7	8	9	10	11	12
25	40	80	36		57	18	46			10	23	63
2	1	1	1		1	2	1			1	2	2

18 找 2 次, 55 找 2 次

3. 设一组初始关键字序列为 (20,18, 22, 16,30, 19,28, 8)，分别用冒泡法排序、插入排序、快速排序（第一个元素 20 为轴元素），给出进行一趟的结果。

冒泡排序 20 18 22 16 30 19 28 8
18 20 16 22 19 28 8 30.

插入排序 18 20 22 16 30 19 28 8

快速排序 20 18 22 16 30 19 28 8
20 18 22 16 8 19 28 30.
20 18 19 16 8 22 28 30

8 18 19 16 20
22 28 30

4. 设一个有序单链表中有 n 个节点，现要求插入一个节点后仍保持有序，操作的时间复杂度为：

$O(n)$

5. 设一个输入序列为 1、2、3、...、n，经过栈的调整，第一个输出元素为 n，则第 i ($1 < i < n$) 个

输出元素为： $n+1-i$

第一个输出为 n 说明前 $n-1$ 个元素都入栈

第二个元素 $n-1$

三 $n-2$

\vdots
i $n-(i-1) = n+1-i$

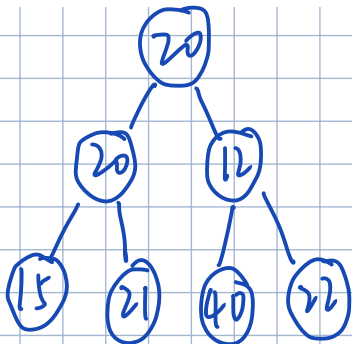
1. 请给出根据二叉树的后序遍历与中序遍历求 二叉树 的算法思想。

① 二叉树的最后一个元素为该树的根，该根在中序中其右侧为该二叉树的左子树中元素，其左侧为该二叉树的右子树中元素。

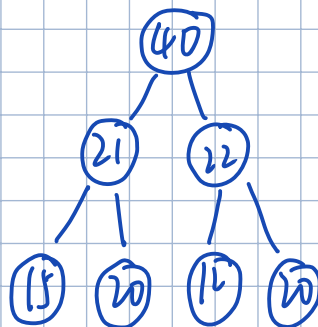
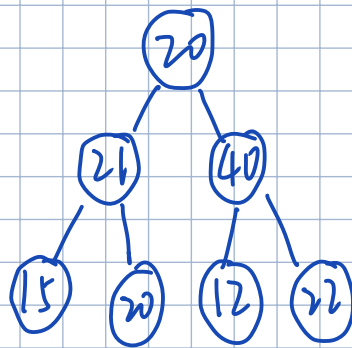
② 根据第一步可将原二叉树的后序和中序遍历划分其左子树的后序和中序遍历 和右子树的后序和中序遍历

③ 依据第二步递归求出该树的结构。

2. 给定一个关键字序列 {20, 20, 12, 15, 21, 40, 22}，请问是否最大堆，若不是，请调整成最大堆。



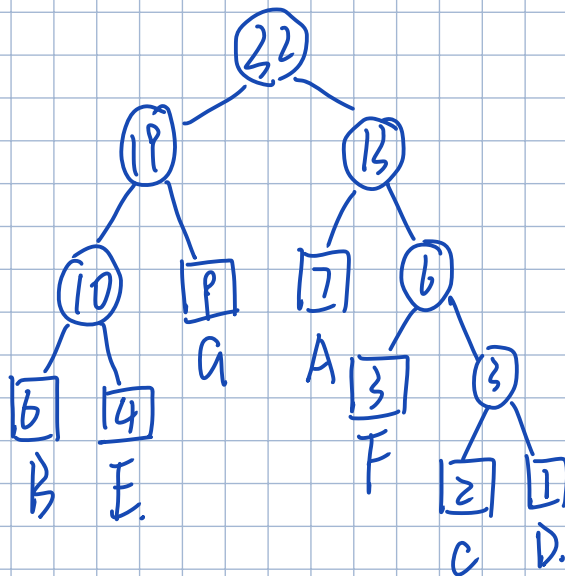
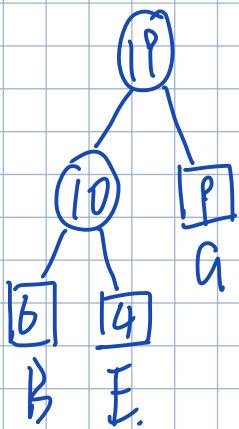
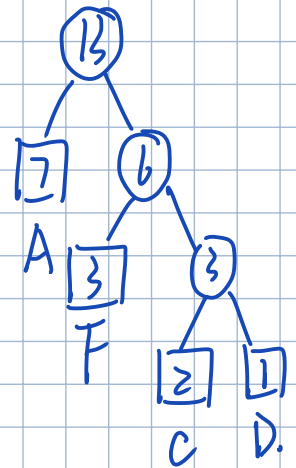
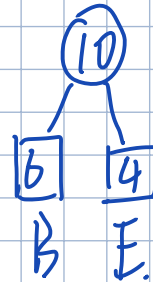
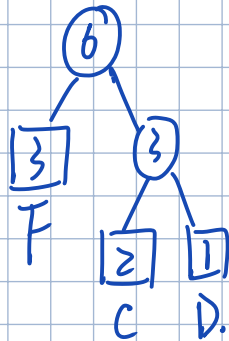
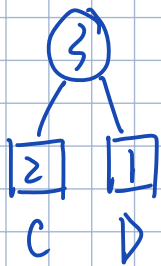
不是最大堆



3. 给定一组数列 (7,6,2,1,4,3,9) 分别代表字符 A,B,C,D,E,F,G 出现的频度, 试叙述建立霍夫曼树的

算法思想, 画出霍夫曼树, 给出各字符的编码值, 并说明这种编码的优点。

A B C D E F G
7 6 2 1 4 3 9

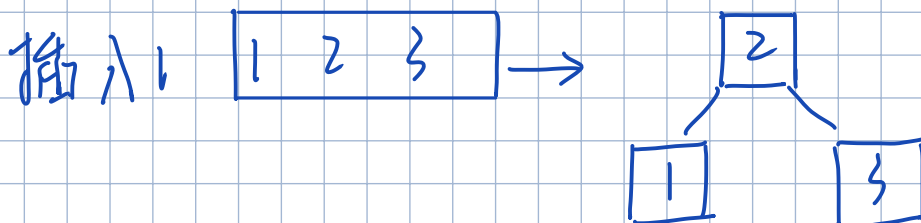


A 10
B 000
C 1110
D 1111
E 001
F 110
G 01

优点: 使用率高的数字编码长度短, 可缩短总体编码长度

4.3 阶 B-树依次插入 3,2,1,4,5,6, 9, 再按照同样的顺序删除。依次画出 B-树的变化。

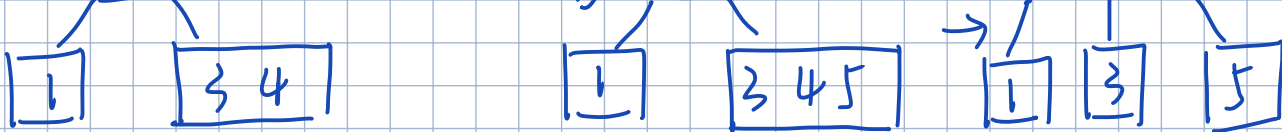
插入3 [3] 插入2 [2 3]



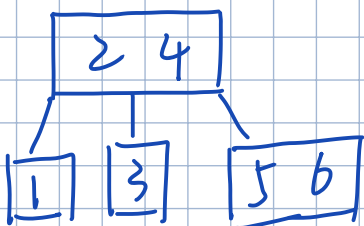
插入4 [2]

插入5 [2]

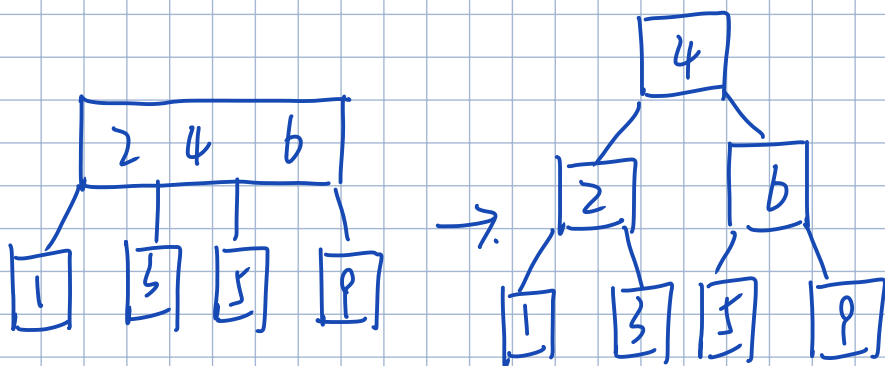
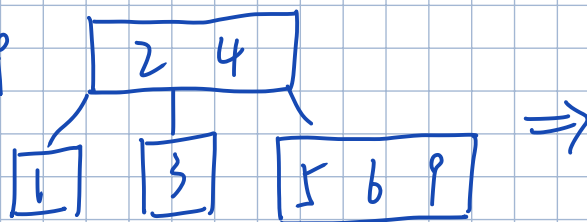
[2 4]



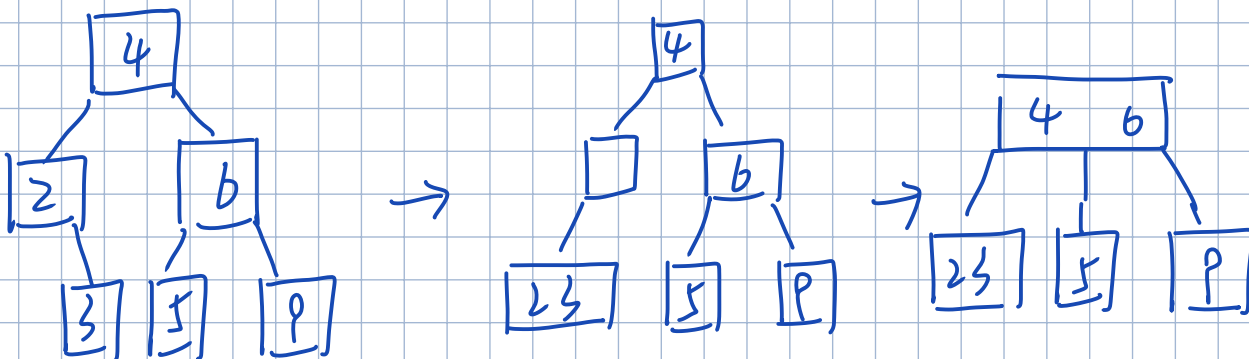
插入 6



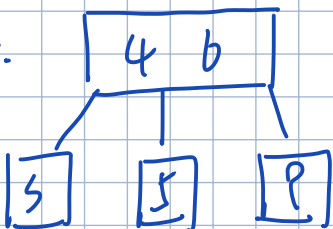
插入 9



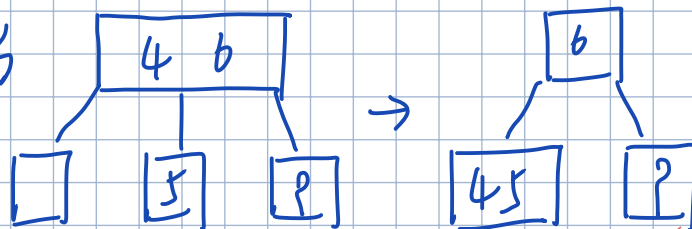
删除 1



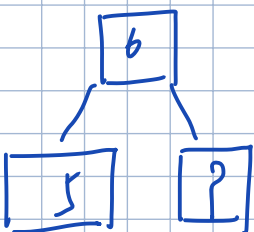
删除 2



删除 3



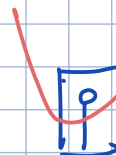
删除 4



删除 5



删除 6

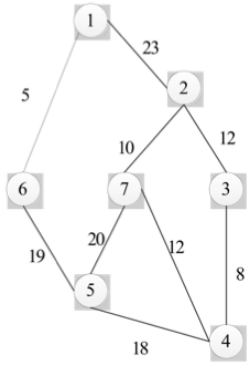


5. 请给出 h 层二叉树的节点数范围。

$$h \sim 2^h - 1$$

(1) 给出 prim 算法的思想

(2) 用 prim (以顶点 5 为起点) 算法计算出如下连通网的最小生成树, 给出计算过程。



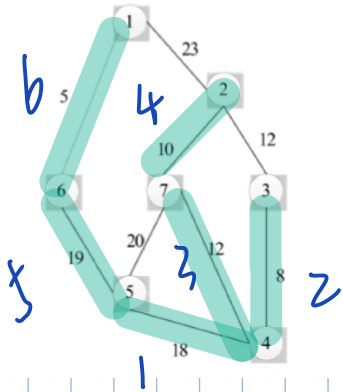
算法思想: 以一个点为起点

将该点加入点集 A

遍历 A 中所有点相邻的

边中权值最小且不与已有边

构成回路的边加入边集 B, 将新边连接的不在点集 A 中顶点加入点集 A, 重复上一步骤直至点集 A 包含图中所有点

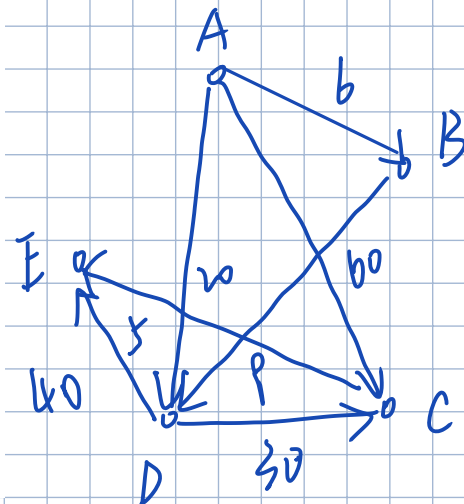


2. (10 分) 下面是某有向加权图(顶点 A,B,C,D,E)的耗费邻接矩阵。

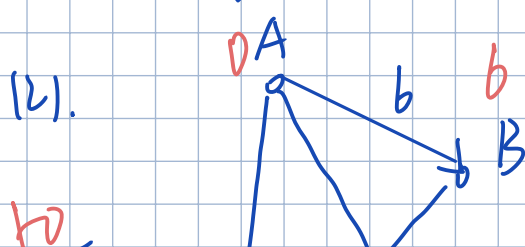
(1) 写出该图的一个拓扑序列,

(2) 使用 Dijkstra 算法依次计算出顶点 A 至其它各顶点的最短路径和最短路径长度。

	A	B	C	D	E
A	0	6	60	20	∞
B	∞	0	∞	9	∞
C	∞	∞	0	∞	5
D	∞	∞	30	0	40
E	∞	∞	∞	∞	0



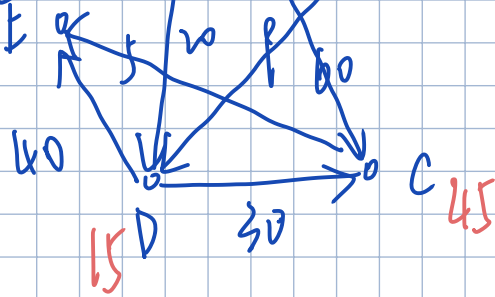
(1) A B D C E



A → B 6

A → B → D 15

A → D → C 40



$$A \rightarrow B \rightarrow D \rightarrow C \rightarrow E = 50$$

3. (5分) 对于 n 个顶点的有向图，用邻接链表表示，简述如何求解下列问题：

(1) 图中有多少条边？

(2) 顶点 i 的入度和出度是多少？

1) 该链表中除头节点外一共有几个链表节点就有多少条边

2) 顶点 i 的入度为除 i 为头节点的链表中其它链表中节点的个数

顶点 i 的出度为以 i 为头节点链表中除头节点外其它节点的数目

1. 已知一个 LinkList 类型的单链表 L (带头结点)，其中每个结点的类型为 LinkNode。LinkNode 中含有两个域：data 域和 next 域，其中 data 域为整型元素，next 域为指针域，指向其直接后继结点。按如下提示的函数原型编写算法，删除 L 中 data 域取值大于 s ，小于 t 的所有结点，说明参数值 $s < t$ 。

```
void DeleteElem( LinkList &L, int s, int t)
```

```
{
}
}
```

```
LinkNode *current = L;
```

```
if (current->next == NULL) {return;}
}
```



```

while (current → next != NULL) {
    int d = current → next → data;
    if (s <= d && d <= t) {
        current → next = current → next → next;
    }
    current = current → next;
}

```

2. 已知有向图 g 采用邻接表的形式存放，有向图中有 n 个顶点，判断顶点 V_i 到 V_j 是否有通路 ($1 \leq i < j \leq n$)，有返回 1，没有返回 0。其中表头结点包含 data 和 firstarc 分别表示数据域和链域，表结点包含 adjvex 和 nextarc 分别表示邻接点域和链域。(注意：本题要求采用非递归算法求解)



```

int count( AdjList g, vertype  $V_i, V_j$  )
{
    ...
}

```

```

Queue<adjvex> q = new Queue<adjvex> ( );
q.push(vi);
while (!q.empty()) {
    adjvex v = q.front();
    q.pop();
    AdjList h = q;
    while (h → data != v → data) { h = h → nextarc; }
}

```

adjvex temp = h → firstarc;

while(temp != null){

if(temp == v_j){

return 1;

}

q.push(temp);

temp = temp → next;

}

}

return 0;

}