

## 第二部分：数据结构

一、填空题（共 5 小题，每小题 2 分，共 10 分）

1. 上三角矩阵的压缩存储方法为\_\_\_\_\_。
2. 已知图  $G(V,E)$ , 其中  $V(G)=\{0,1,2,3\}$ ,  $E(G)=\{(0,1),(0,2),(0,3),(1,3),(2,3)\}$ 。对该图从顶点 0 开始分别进行深度优先遍历和广度优先遍历所得的序列各有\_\_\_\_\_、\_\_\_\_\_种。
3. 如图 1 所示的树中：若树中每个结点的存储需要占用 4 字节，则采用顺序存储该树需要\_\_\_\_\_字节的空间，其广义表表示为\_\_\_\_\_。
4. 折半查找表中的元素值一定是\_\_\_\_\_的，但索引顺序查找表中仅要求\_\_\_\_\_有序。
5. 归并排序\_\_\_\_\_（填是或者不是）稳定的，其时间复杂度为\_\_\_\_\_。

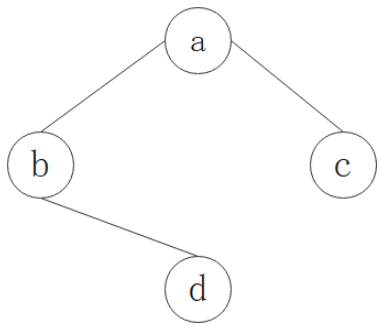


图 1

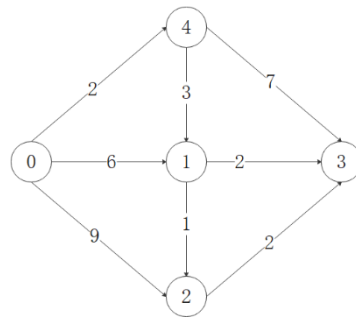


图 2

二、单选题（共 10 小题，每题 2 分，共 20 分）

1. 下列关于算法的叙述中，正确的是( )。  
A. 算法执行时间只跟数据规模  $n$  有关  
B. 算法分析的目的是研究算法的输入和输出的关系  
C. 算法只能用流程图表示  
D. 算法特性有确定性、可读性、有限性、有输入、有输出
2. 一个栈的入栈序列为 a,b,c,d,e,f, 则不可能的输出序列为( )。  
A. cbfed  
B. badfec  
C. dfebac  
D. decbfa
3. 下列有关图的相关算法的叙述中，正确的是( )。  
A. Prim 算法和 Kruskal 算法得到的生成树一定一致  
B. Prim 算法适合边稀疏的图  
C. Kruskal 算法适合边稠密的图  
D. Kruskal 算法的时间复杂度为  $O(e \log e)$
4. 对关键字序列 {56,23,78,92,88,67,19,34} 进行增量为 3 的一趟希尔排序的结果为( )。  
A. {19,23,56,34,78,67,88,92}  
B. {23,56,78,66,88,92,19,34}  
C. {19,23,34,56,67,78,88,92}  
D. {19,23,67,56,34,78,92,88}

5. 一棵高度为  $h$  的二叉树所含结点最多及第  $h$  层结点数最多为( )。  
A.  $2^h - 1, 2^{h-1}$   
B.  $2^{h-1}, 2^h$   
C.  $2h, 2h - 1$   
D.  $2^h, 2^{h-1}$
6. 下列选项中，不能保证第一趟排序完成后将一个元素放在最终位置的是( )。  
A. 希尔排序  
B. 选择排序  
C. 冒泡排序  
D. 快速排序
7. 折半查找有序表 {2,10,25,35,40,65,70,75,81,82,88,100}，若查找元素 75，需要依次与表中元素( )进行比较。  
A. 65, 82, 75  
B. 70, 82, 75  
C. 65, 81, 75  
D. 65, 81, 70, 75
8. 使用迪杰斯特拉 (Dijkstra) 算法求图 2 中从顶点 0 到其他各顶点的最短路径，依次得到的各最短路径的目标顶点是( )。  
A. {0,4,2,3,1}  
B. {0,4,2,1,3}  
C. {0,4,1,2,3}  
D. {0,4,1,2,3}
9. 在一个单链表中，若  $p$  所指的结点不是最后一个结点，在  $p$  结点之后插入  $s$  所指的结点，则执行( )。  
A.  $s->next = p; p->next = s;$   
B.  $p->next = s; s->next = p;$   
C.  $p = s; s->next = p->next;$   
D.  $s->next = p->next; p->next = s;$
10. 已知有向图  $G(V,E)$ , 其中  $V(G)=\{1, 2, 3, 4, 5, 6, 7\}$ ,  $E=\{(1,2),(1,3),(1,4),(2,5),(3,5),(3,6),(4,6),(5,7),(6,7)\}$ ,  $G$  的拓扑排序序列可以是( )。  
A. {1, 3, 4, 6, 2, 5, 7}  
B. {1, 3, 2, 6, 4, 5, 7}  
C. {1, 3, 4, 5, 2, 6, 7}  
D. {1, 2, 5, 3, 4, 6, 7}

三、简答题（共 6 小题，每小题 5 分，共 30 分）

1. 试用链式存储结构实现队列，要求入队和出队时间复杂度尽可能低。
2. 如果一个递归函数可以表达为  $f(n) = 2f(n/2) + O(n)$ ，计算  $f(n)$  时间复杂度。
3. 已知序列 {6, 5, 2, 4, 3, 1}，请按输入序构造平衡二叉树。
4. 给定关键字序列 {30,25,20,15,10}，试将其调整为大根堆，再给出每一趟非递减堆排序的结果。
5. 已知输入关键字序列为 {7, 1, 4, 9, 10, 2, 6, 5}，根据哈希函数建立哈希表。已知哈希函数为  $\text{Hash}(\text{key}) = \text{key} \bmod 11$ ，哈希表长为 11，溢出表长为 3。请画出哈希表和溢出表，并计算查找成功时(等概率情况下)的平均查找长度 ASL。
6. 全源最短路径问题采用 Floyd 算法进行求解。下面给出了一个由 4 个顶点构成的有向图邻接矩阵  $\text{Dist}[4][4]$  和路径矩阵  $\text{Path}[4][4]$ 。约定  $\text{Dist}$  中用  $\infty$  表示不能到达， $\text{Path}$  中用 -1 表示没有前驱的情况。请计算并给出每一次迭代的结果。

	Dist <sup>(-1)</sup>				Dist <sup>(0)</sup>				Dist <sup>(1)</sup>				Dist <sup>(2)</sup>				Dist <sup>(3)</sup>			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
0	0	3	∞	8																
1	3	0	5	∞																
2	7	∞	0	1																
3	5	∞	∞	0																

	Path <sup>(-1)</sup>				Path <sup>(0)</sup>				Path <sup>(1)</sup>				Path <sup>(2)</sup>				Path <sup>(3)</sup>			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
0	-1	0	-1	0																
1	1	-1	1	-1																
2	2	-1	-1	2																
3	3	-1	-1	-1																

四、算法设计题（共 15 分）

1. (7 分)给定 n 个元素的数组, 求出给出其中的任意元素与其后任意元素差值的最大值(即 max(array[i]-array[j]) （0<=i<j<n), 时间复杂度和空间复杂度尽可能低。

参考函数原型:int GetMax(int \*array)

2. (8 分)所谓镜像二叉树就是以根结点做铅垂线，左右子树关于该铅垂线对称。试设计一个算法判断给定二叉树是否为镜像二叉树。其中二叉树结点定义为：

struct BiTreeNode{int data;struct BiTreeNode \*left;struct BiTreeNode \*right};

参考函数原型： int IsSymmetricTree(BiTreeNode \*root)