

课程名称: **数据结构与算法 (B)**

2018-2019 学年第 (2) 学期期末

本试卷共 4 道大题, 满分 100 分

(考试结束后请将试卷、答题本一起交给监考老师)

一、选择题 (每空 3 分, 共 30 分)

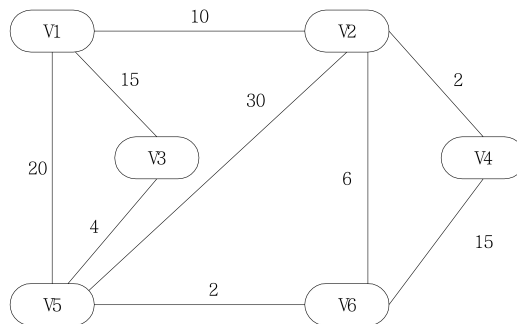
- 1、对于数据的存储结构来说, 常见的存储映射方法分为顺序结构、链式结构、索引结构和 ()。
A. 线性结构; B. 散列结构; C. 树结构; D. 图结构;
- 2、设有三个元素 X, Y, Z 顺序进栈 (进的过程中允许出栈), 下列得不到的出栈排列是()。
A. XYZ B. YZX C. ZXY D. ZYX
- 3、在下述结论中, 正确的是 ()。
①只有一个结点的二叉树的度为 0; ②二叉树是度数为 2 的树; ③二叉树的左右子树可任意交换;
④深度为 K 的完全二叉树的结点个数小于或等于深度相同的满二叉树。
A. ①②④ B. ①④ C. ②④ D. ①
- 4、一棵二叉树的先根遍历结果是 EFHIGJK, 中根遍历结果是 HFIEJKG, 该二叉树根的右子结点是 ()。
A、 E B、 F C、 G D、 H
- 5、表达式 $a*(b+c)-d$ 的后缀表达式是()。
A. $abcd*+-$ B. $abc+*d-$ C. $abc*+d-$ D. $-+*abcd$
- 6、设散列表长 $m=12$, 散列函数 $H(\text{Key})=\text{Key} \bmod 11$, 表中已存储 3 个结点, $\text{addr}(15)=4$, $\text{addr}(38)=5$, $\text{addr}(84)=7$, 其余地址为空。如果双散列函数 $H_2(\text{Key})=\text{Key} \bmod 3$ 解决碰撞, 则关键字为 49 的元素的实际存储地址为 ()。
A. 5 B. 6 C. 8 D. 9
- 7、内排序方法的稳定性是指()。
A. 该排序算法不允许有相同的关键字记录 B. 该排序算法允许有相同的关键字记录
C. 平均时间为 $O(n \log n)$ 的排序方法 D. 以上都不对
- 8、用最快的速度从 1000 个无序元素挑选出其中前十个最大的元素, 采用以下 () 排序方法较好。
A. 快速排序 B. 归并排序 C. 堆排序 D. 基数排序
- 9、下列哪一种图的邻接矩阵是对称矩阵? ()
A. 无向图 B. 有向图 C. AOV 网 D. AOE 网
- 10、一个有 n 个结点的图, 最少有 () 个连通分量。
A. 0 B. 1 C. $n-1$ D. n

二、填空题 (每空 3 分, 共 30 分)

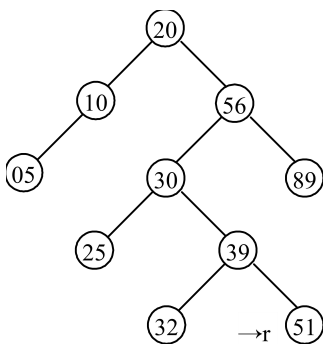
- 1、设循环队列存放在顺序表 $sq.data[M]$ 中，初始队头指针 $sq.front$ 和队尾指针 $sq.rear$ 都等于 0，则出队操作可表示为_____，若用牺牲一个单元的办法来区分队满和队空，则队满的条件为_____。
- 2、设字符串正文串长度为 n ，模式串长度为 m ，则串匹配的 KMP 算法的时间复杂度为_____。
- 3、深度为 K （树中层数最大的叶结点的层数）的完全二叉树至少有_____个叶结点，至多有_____个叶结点。
- 4、对一组记录进行非递减排序，其关键码为 (46, 70, 56, 38, 40, 80)，则利用快速排序的方法，以第一个记录为基准得到的第一次划分结果为 (_____); 采用直接插入排序方法，第一次扫描后的结果是 (_____); 采用初始步长为 3 的希尔 (shell) 排序，第一趟扫描的结果是 (_____); 而采用归并排序第一次扫描后的结果是 (_____)
- 5、已知有向图 $G=(V,E)$ ，其中 $V=\{V_1, V_2, V_3, V_4, V_5, V_6\}$ ， $E=\{<V_1, V_2>, <V_1, V_3>, <V_1, V_4>, <V_2, V_5>, <V_3, V_5>, <V_3, V_6>, <V_4, V_6>, <V_4, V_2>\}$ ， G 的拓扑序列是_____。

三、简答题 (共 20 分)

- 1、对于下图 G ，画出其邻接表，给出从顶点①开始的深度周游生成树和广度周游生成树，再利用基于边的 $kruskal$ 算法产生图的最小生成树，给出 mst 数组的变化。(12 分)



- 2、已知有如下二叉搜索（排序）树，请给出删除数据为 56 节点之后的两种调整规则 and 结果。(8 分)



四、算法设计 (共 20 分)

1、请完成下列 Dijkstra 算法。(12 分)

```
#define MAX 1e+38
void dijkstra(Graph graph, Path dist[ ] )
{
    int i, j, minvex;
    float min;
    dist[0].length=0; dist[0].prevex=0;
    dist[0].vertex=graph.vexs[0];
    graph.arcs[0][0]=1;          /* 表示顶点 v0 在集合 U 中 */
    for(i=1; i<graph.n; i++)      /* 初始化集合 V-U 中顶点的距离值 */
    {
        dist[i].length=graph.arcs[0][i];
        dist[i].vertex=graph.vexs[i];
        if(dist[i].length!=MAX)    dist[i].prevex=0;
        else dist[i].prevex= -1;
    }
    for(i=1; i<graph.n; i++)
    {
        min=MAX;    minvex=0;
        for(j=1; j<graph.n; j++)      /*在 V-U 中选出距离值最小顶点*/
            if( (graph.arcs[j][j]==0) && (_____ ) )
                {
                    min=dist[j].length;
                    minvex=j;
                }
        if(_____)    break;          /* 从 v0 没有路径可通往集合 V-U 中的顶点 */
        graph.arcs[minvex][minvex]=1; /* V-U 路径最小顶点 minvex */
        for(j=1; j<graph.n; j++)      /* 调整集合 V-U 中的顶点的最短路径 */
        {
            if(graph.arcs[j][j]==1) continue;
            if(dist[j].length>dist[minvex].length+graph.arcs[minvex][j])
            {
                _____;
                _____;
            }
        }
    }
}
```

2、设 float element[N] (N 为正整数常数) 存储一个班的数学考试成绩, 请编写一个算法, 用尽可能小的时间复杂性, 将数组中的元素分为三类: 一类是大于等于 85 分优秀, 一类是小于 85 分大于等于 60 分良好, 另一类是小于 60 分不及格, 并按照优秀、良好、不及格的顺序存储, 并返回第二类 and 第三类元素的起始下标。(8 分)

```
// beginGood 返回第一个成绩为良好的记录的下标
// beginFail 返回第一个成为为不及格的记录的下标
// int n 为数组 element 中的有效记录数目
int ScoreClassify ( float *element, int n, int * beginGood, int beginFail)
{

}

}
```