

# 2021 天勤计算机考研八套模拟卷 · 卷五

## 数据结构篇

### 一、选择题

1. 在  $n$  个结点的线性表的数组实现中，算法的时间复杂度是  $O(1)$  的操作是（ ）。
- A. 访问第  $i$  ( $1 \leq i \leq n$ ) 个结点和求第  $i$  个结点的直接前驱 ( $2 \leq i \leq n$ )
- B. 在第  $i$  ( $1 \leq i \leq n$ ) 个结点后插入一个新结点
- C. 删除第  $i$  ( $1 \leq i \leq n$ ) 个结点
- D. 以上都不对
2. 以下说法正确的是（ ）。
- I. 带头结点的循环双链表  $L$  为空的条件是:  $L \rightarrow \text{prior} == L \& \& L \rightarrow \text{next} == L$
- II. 线性表的插入和删除总是伴随着大量数据的移动
- III. 只有删除静态链表的尾结点才不需要移动元素
- IV. 若线性表采用链式存储结构，要求内存中可用存储单元的地址必须不连续
- A. 仅 I
- B. 仅 I、II
- C. 仅 II、III
- D. I、II、III 和 IV
3. 循环队列用数组  $A[0 \cdots m-1]$  存放其元素值，已知其头尾指针分别是  $\text{front}$  和  $\text{rear}$ （且队尾指针  $\text{rear}$  指向队尾元素的下一个元素），则当前队列中的元素个数是（ ）。
- A.  $(\text{rear} - \text{front} + m) \% m$
- B.  $(\text{rear} - \text{front} + 1) \% m$
- C.  $\text{rear} - \text{front} - 1$
- D.  $\text{rear} - \text{front}$
4. 下列关于二叉树的叙述中正确的是（ ）。
- I. 对于任何一棵二叉树，叶子结点数都是度为 2 的结点数加 1
- II. 二叉树的左右子树不可以任意地交换
- III. 二叉树只适合使用链式结构存储，不可能用顺序结构存储
- IV. 结点按层序编号的二叉树，第  $i$  个结点的左孩子（假设存在）的编号为  $2i$
- A. 仅 I、II
- B. 仅 II
- C. 仅 II、IV
- D. 仅 II、III
5. 若二叉树是由森林变换而来的，若森林中有  $n$  个非终端结点，则二叉树中无右孩子的结点有（ ）。
- A.  $n-1$
- B.  $n$
- C.  $n+1$
- D.  $n+2$
6. 已知二叉树后序遍历序列是  $\text{dabec}$ ，中序遍历序列是  $\text{debac}$ ，它的前序遍历序列是（ ）。
- A.  $\text{acbed}$
- B.  $\text{decab}$
- C.  $\text{deabc}$
- D.  $\text{cedba}$



2. 给定一字符串，该字符串中存在若干对相同的字符，设计一个在时间和空间上尽可能高效的算法，找出一对相同字符在该字符串中的最大距离。例如：“KLabcLdecL”，其中第一个“L”和最后一个“L”相距最远，它们在原字符串中的位置相差 8，要求：

- (1) 给出算法的基本设计思想。
- (2) 根据设计思想，采用 C、C++ 语言描述算法，关键之处给出注释。
- (3) 说明你所设计算法的时间复杂度和空间复杂度。

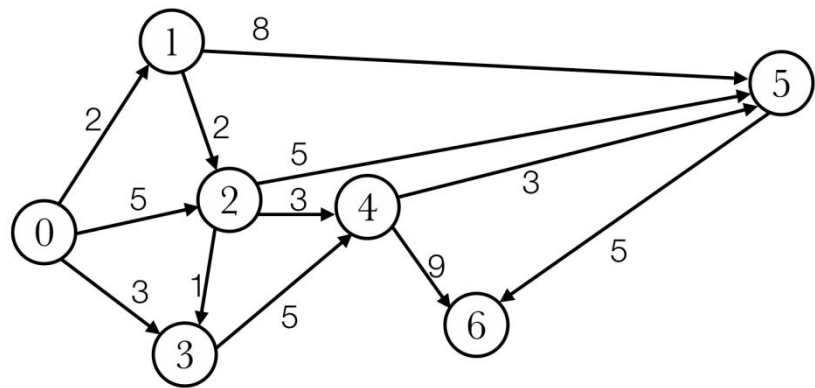
答案

一、选择题答案

1.A    2.A    3.A    4.B    5.C    6.D    7.A    8.C    9.D    10.B

二、综合题答案

1.  
(1) 有向图:



(2) 邻接表

0	→	1	2	→	2	5	→	3	3	^
1	→	2	2	→	5	8	→	^		
2	→	3	1	→	4	3	→	5	5	^
3	→	4	5	→	^					
4	→	5	3	→	6	9	→	^		
5	→	6	5	→	^					
6	→	^								

(3)  
0 到其余顶点到最短路径长度:

	0	1	2	3	4	5	6
dist	0	2	4	3	7	9	14

2.  
(1) 设计思想:

(2) 在遍历字符数组的过程中，对已经访问过的字符进行标记，同时记下它们第一次出现在原字符串中的位置，当以后再次遍历到此字符时，根据当前的位置和第一次出现的位置，求出它们之间的距离，然后用得到的距离和当前最大距离相比较。为此需要设置一个数组用来存放已经访问过的字符，为了能加快搜索，采用字符的 ASCII 码作为数组的下标来支持随机访问，通过对应下标中的数组元素的访问标记来判断它是否被访问过，同时获取它第一

次出现的位置，每遍历一个字符就开始判断，并用 max 来保存当前最大距离。

代码:

```
int getMaxLength(char str[],int n)
{
    int max=0;
    int i;
    int isChInArray[128];
    for(i=0; i<128; ++i)
        isChInArray[i] = -1;
    for(i=0; i<n; ++i)
    {
        if(isChInArray[str[i]] == -1)
            isChInArray[str[i]] = i;
        else
        {
            int tempL = i - isChInArray[str[i]];
            if(max < tempL)
                max = tempL;
        }
    }
    return max;
}
```

时间复杂度  $O(n)$ 。

空间复杂度  $O(1)$ 。

全套模拟卷以及答案解析视频讲解来辉解读公众号获取:

