2021 天勤计算机考研八套模拟卷 • 卷三

数据结构篇

一、选择题（单选）

1. 在双链表中 p 所指的结点之前插入一个结点 q 的操作为（　　）。
A. p→prior=q; q→next=p; p→prior→next=q; q→prior=p→prior;
B. q→prior=p→prior; p→prior→next=q; q→next=p; p→prior=q→next;
C. q→next=p; p→next=q; q→prior→next=q; q→next=p;
D. p→prior→next=q; q→next=p; q→prior=p→prior; p→prior=q;

2. 下列关于链式栈的叙述中，错误的是（　　）。
Ⅰ. 链式栈只能顺序存取，而顺序栈不但能顺序存取，还能直接存取
Ⅱ. 因为链式栈没有栈满问题，所以进行进栈操作，不需要判断任何条件
Ⅲ. 在链式队列的出队操作中，需要修改尾指针的情况发生在出队操作后队空的时候
A. 仅Ⅰ　　B. 仅Ⅰ、Ⅱ
C. 仅Ⅱ　　D. Ⅰ、Ⅱ、Ⅲ

3. 设有一个二维数组 A[m][n]在存储中按行优先存放（数组的每一个元素占一个空间），假设 A[0][0]存放位置在 780，A[4][6]存放位置在 1146，则 A[6][20]在（　　）位置。
A. 1342　　B. 1336
C. 1338　　D. 1340

4. 一棵二叉树的前序遍历序列为 1234567，则它的中序遍历序列不可能是（　　）。
Ⅰ. 3124567　　Ⅱ. 1234567
Ⅲ. 4135627　　Ⅳ. 1436572
A. 仅Ⅰ、Ⅱ　　B. 仅Ⅱ、Ⅲ
C. 仅Ⅰ、Ⅲ　　D. 仅Ⅰ、Ⅲ、Ⅳ

5. 宽度为 27，高度为 4 的满 N 叉树总共有（　　）个结点。
A. 27　　B. 40
C. 85　　D. 97

6. 对于一棵具有 n 个结点、度为 4 的树来说（树的层数从 1 开始），以下说法正确的是（　　）。
Ⅰ. 树的高度至多为 n-3
Ⅱ. 至少在某一层上正好有 4 个结点
Ⅲ. 第 i 层上至多有 4（i-1）个结点
A. 仅Ⅰ　　B. 仅Ⅰ、Ⅱ
C. 仅Ⅱ　　D. 仅Ⅰ、Ⅲ

7. 以下有关拓扑排序的说法中，错误的是（　　）。

Ⅰ. 如果某有向图存在环路，则该有向图一定不存在拓扑排序

Ⅱ. 在拓扑排序算法中，既可以使用栈，也可以使用队列

Ⅲ. 若有向图的拓扑有序序列唯一，则图中每个顶点的入度和出度最多为 1

A. 仅Ⅰ、Ⅲ　　B. 仅Ⅱ、Ⅲ

C. 仅Ⅱ　　D. 仅Ⅲ

8. 无向图 G 有 23 条边，度为 4 的顶点有 5 个，度为 3 的顶点有 4 个，其余都是度为 2 的顶点，则图 G 最多有（　　）个顶点。

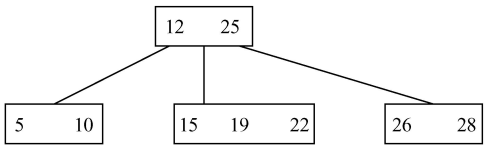
A. 11　　B. 12

C. 15　　D. 16

9. 下图是一棵（　　）。

A. 4 阶 B-树　　B. 4 阶 B+树

C. 3 阶 B-树　　D. 3 阶 B+树



题 9 图

10. 如果一台计算机具有多个可并行运行的 CPU，就可以同时执行相互独立的任务。归并排序的各个归并段的归并也可并行执行，因此称归并排序是可并行执行的。那么以下的排序方法不可以并行执行的有（　　）。

Ⅰ. 基数排序　　Ⅱ. 快速排序

Ⅲ. 起泡排序　　Ⅳ. 堆排序

A. 仅Ⅰ、Ⅲ　　B. 仅Ⅰ、Ⅱ

C. 仅Ⅰ、Ⅲ、Ⅳ　　D. 仅Ⅱ、Ⅳ

二、综合题

1. 分别用递归和非递归的方法实现将一个正整数字符串转化为对应的整数。

2. 设以二元组（f，c）的形式保存一棵树的各条边（其中 f 是双亲结点标识，c 是孩子结点标识），f 为'^'时 c 表示根结点，f 和 c 都为'^'时，输入结束。编写一个算法，由给定的二元组序列（已保存在字符串数组 tTuple[maxSize][3]中）建立这棵树的孩子—兄弟链表存储结构。

注意：双亲结点相同的二元组在序列中一定连续出现。

答案

一、选择题答案

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

二、综合题答案

1.

非递归:

```
int stringToInt(char* ch)
{
  int sum = 0;
  int i = 0;
  int forExp = 1;
  while(ch[i] != '\0') ++i;
  --i;

  while(i >= 0)
  {
    sum += (ch[i]-'0') * forExp;
    forExp*=10;
    --i;
  }
  return sum;
}
```

递归:

```
int stringToIntRecursion(char *ch, int L, int R)
//把下标 L 和 R 以及其之间的字符串转化为整数
{
  if (L > R)
    return -1;
  if (L == R)
    return ch[R] - '0';
  else
    return stringToIntRecursion(ch, L, R-1)*10 + ch[R] - '0';
}
```

2.

```
CSBTNode* createTree(char tTuple[][3])
{
  CSBTNode* t = NULL;
  CSBTNode* parents[maxSize]; int pn = 0;
  int i = 0;
  while(tTuple[i][0] != '^' || tTuple[i][1] != '^')
  {
    if(tTuple[i][0] =='^')
    {
      t = (CSBTNode*)malloc(sizeof(CSBTNode));
      t->data = tTuple[i][1];
      t->child = t->sibling = NULL;
      parents[pn] = t;
      ++pn;
      ++i;
    }
    else
    {
      CSBTNode* tempP = NULL;
      CSBTNode* lS = NULL;
      int j=0;
      for(j=0; j<pn; ++j)
        if (tTuple[i][0] == parents[j]->data)
          tempP = parents[j];
      int flag = 0;
      for(j=i; tTuple[j][0] == tempP->data; ++j)
      {
        if(flag == 0)
        {
          flag = 1;
          CSBTNode* tempC = (CSBTNode*)malloc(sizeof(CSBTNode));
          tempC->data = tTuple[j][1];
          tempC->child = tempC->sibling = NULL;
          tempP->child = tempC;
          lS = tempC;
          parents[pn] = tempC;
          ++pn;
        }
        else
        {
          CSBTNode* tempS = (CSBTNode*)malloc(sizeof(CSBTNode));
          tempS->data = tTuple[j][1];
          tempS->child = tempS->sibling = NULL;
          lS->sibling = tempS;
          lS = tempS;
          parents[pn] = tempS;
```

```
            ++pn;
          }
        }
        i = j;
      }
    }
    return t;
}
```

全套模拟卷以及答案解析视频讲解来辉解读公众号获取：