

每日一题Java方向day19_6月22日

一. 单选

1. 下列关于线性链表的叙述中，正确的是（ ）。

- ☐ A 各数据结点的存储空间可以不连续，但它们的存储顺序与逻辑顺序必须一致
- ☐ B 各数据结点的存储顺序与逻辑顺序可以不一致，但它们的存储空间必须连续
- ☐ C 进行插入与删除时，不需要移动表中的元素
- ☐ D 以上说法均不正确

正确答案：C

2.

一个栈的初始状态为空。现将元素 1,2,3,A,B,C 依次入栈，然后再依次出栈，则元素出栈的顺序是（ ）

- ☐ A 1,2,3,A,B,C
- ☐ B C,B,A,1,2,3
- ☐ C C,B,A,3,2,1
- ☐ D 1,2,3,C,B,A

正确答案：C

3. 用不带头结点的单链表存储队列,其队头指针指向队头结点,队尾指针指向队尾结点,则在进行出队操作时()

- ☐ A 仅修改队头指针
- ☐ B 仅修改队尾指针
- ☐ C 队头、队尾指针都可能要修改
- ☐ D 队头、队尾指针都要修改

正确答案：C

4. 递归函数最终会结束，那么这个函数一定？

- ☐ A 使用了局部变量
- ☐ B 有一个分支不调用自身
- ☐ C 使用了全局变量或者使用了一个或多个参数
- ☐ D 没有循环调用

正确答案：B

5. 已知二叉树后序遍历序列是bfegcda，中序遍历序列是badefcg，它的前序遍历序列是：

- A abcdefg
- B abdcefg
- C adbcfeg
- D abecdfg

正确答案：B

6.

某完全二叉树按层次输出（同一层从左到右）的序列为 ABCDEFGH。该完全二叉树的前序序列为（ ）

- A ABDHECFG
- B ABCDEFGH
- C HDBEAFCG
- D HDEBFGCA

正确答案：A

7. 以下序列不是堆的是()

- A (100,85,98,77,80,60,82,40,20,10,66)
- B (100,98,85,82,80,77,66,60,40,20,10)
- C (10,20,40,60,66,77,80,82,85,98,100)
- D (100,85,40,77,80,60,66,98,82,10,20)

正确答案：D

8. 设有一组记录的关键字为{19,14,23,1,68,20,84,27,55,11,10,79},用链地址法构造哈希表,哈希函数为 $H(key)=key \text{ MOD } 13$,哈希地址为1的链中有()个记录

- A 1
- B 2
- C 3
- D 4

正确答案：D

9.

假设你只有100Mb的内存，需要对1Gb的数据进行排序，最合适的算法是？

- A 归并排序
- B 插入排序
- C 快速排序
- D 冒泡排序

正确答案：A

10.

下列哪种图的邻接矩阵是对称矩阵（ ）。

- A 有向图
- B 无向图
- C AOV图
- D AOE图

正确答案：B

二. 编程

1. 标题：子串判断 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

现有一个小写英文字母组成的字符串s和一个包含较短小写英文字符串的数组p，请设计一个高效算法，对于p中的每一个较短字符串，判断其是否为s的子串。

给定一个string数组p和它的大小n，同时给定string s，为母串，请返回一个bool数组，每个元素代表p中的对应字符串是否为s的子串。保证p中的串长度小于等于8，且p中的串的个数小于等于500，同时保证s的长度小于等于1000。

测试样例：

["a","b","c","d"],4,"abc"

返回：[true,true,true,false]

输入描述：

输出描述：

示例1:

输入

输出

正确答案：

2. 标题：成绩排序 | 时间限制：1秒 | 内存限制：32768K

查找和排序

题目：输入任意（用户，成绩）序列，可以获得成绩从高到低或从低到高的排列,相同成绩都按先录入排列在前的规则处理。

例示：

```
jack 70
peter 96
Tom 70
smith 67
```

从高到低 成绩

```
peter 96
jack 70
Tom 70
smith 67
```

从低到高

```
smith 67

Tom 70
jack 70
peter 96
```

输入描述：

输入多行，先输入要排序的人的个数，然后分别输入他们的名字和成绩，以一个空格隔开

输出描述：

按照指定方式输出名字和成绩，名字和成绩之间以一个空格隔开

示例1:

输入

3

0

fang 90

yang 50

ning 70

输出

fang 90

ning 70

yang 50

正确答案：