



1、引入二叉线索树的目的是（ ）。(点击查看答案>>>>>>)

- A、加快查找结点的前驱或后继的速度
- B、为了能在二叉树中方便的进行插入与删除
- C、为了能方便的找到双亲
- D、使二叉树的遍历结果唯一

难度：★★

- 2、假设以行优先顺序存储三维数组 $A[5][6][7]$, 其中元素 $A[0][0][0]$ 的地址为 1100, 且每个元素占 2 个存储单元, 则 $A[4][3][2]$ 的地址是 ()

(点击查看答案>>>>>>>)

- A、 1150
B、 1291
C、 1380
D、 1482

难度: ★★

- 3、在栈中存取数据的原则是_____。(点击查看答案>>>>>>>)

- A、先进先出
B、先进后出
C、后进后出
D、随意进出

难度: ★★

- 4、一棵有 124 个叶结点的完全二叉树，最多有 () 个结点。

(点击查看答案>>>>>>>)

- A、 247
B、 248
C、 249
D、 251

难度: ★★



5、某二叉树的后序遍历序列与中序遍历序列相同，均为 ABCDEF，则按层次输出（同一层从左到右）的序列为（ ）[（点击查看答案>>>>>>>>）](#)

- A、FEDCBA
- B、CBAFED
- C、DEFCBA
- D、ABCDEF

难度：★★★

6、双向链表中有两个指针域，llink 和 rlink，分别指回前驱和后继，设 p 指向链表中的一个结点，q 指向一待插入结点，现要求在 p 前插入 q，则正确的插入为（ ）[（点击查看答案>>>>>>>>）](#)

- A、p->llink=q;q->rlink=p;p->llink->rlink=q;q->llink:=p->llink;
- B、q->llink=p->llink;p->llink->rlink=q;q->rlink=p;p->llink=q->rlink;
- C、q->rlink=p;p->rlink:=q;p->llink->rlink=q;q->rlink=p;
- D、p->llink->rlink=q;q->rlink=p;q->llink=p->llink;p->llink=q;

难度：★★★

7、将森林 F 转换为对应的二叉树 T，F 中叶结点的个数等于（ ）[（点击查看答案>>>>>>>>）](#)

- A、T 中叶结点的个数
- B、T 中度为 1 的结点个数
- C、T 中左孩子指针为空的结点个数
- D、T 中右孩子指针为空的结点个数

难度：★★★

8、以下不合法的数组定义是（ ）[（点击查看答案>>>>>>>>）](#)

- A、char a[] = "good"
- B、int s[5] = {1, 2, 3, 4};
- C、char s = "good";
- D、char s[] = {0, 1, 2, 3, 4};

难度：★★★★



9、设有一个 10 阶对称矩阵 $A[10][10]$ ，采用压缩存储方式按行将矩阵中的下三角部分的元素存入一维数组 $B[]$ 中， $A[0][0]$ 存入 $B[0]$ 中，则 $A[8][6]$ 在 $B[]$ 的 () 位置。[\(点击查看答案>>>>>>>>\)](#)

- A、32
- B、33
- C、42
- D、65

难度：★★★★

10、java 中，StringBuilder 和 StringBuffer 的区别，下面说法错误的是？[\(点击查看答案>>>>>>>>\)](#)

- A、StringBuffer 是线程安全的
- B、StringBuilder 是非线程安全的
- C、StringBuffer 对 String 类型进行改变的时候其实都等同于生成了一个新的 String 对象，然后将指针指向新的 String 对象。
- D、效率比较 String<StringBuffer<StringBuilder，但是在 String S1 = “This is only a” + “ simple” + “ test” 时，String 效率最高。

难度：★★★★

11、若用邻接矩阵存储有向图，矩阵中主对角线以下的元素均为零，则关于该图拓扑序列的结论是 ()。[\(点击查看答案>>>>>>>>\)](#)

- A、存在，且唯一
- B、存在，且不唯一
- C、存在，可能不唯一
- D、无法确定是否存在

难度：★★★★

12、具有 7 个顶点的有向图至少应有多少条边才可能成为一个强连通图？

[\(点击查看答案>>>>>>>>\)](#)

- A、6
- B、7
- C、8
- D、12

难度：★★★★



13、一棵二叉树的后序遍历序列为 EFHIGJK，中序遍历序列为 HFIEJKG，则该二叉树根结点的右孩子为（ ）[\(点击查看答案>>>>>>>>\)](#)

- A、E
- B、F
- C、G
- D、H

难度：★★★★★

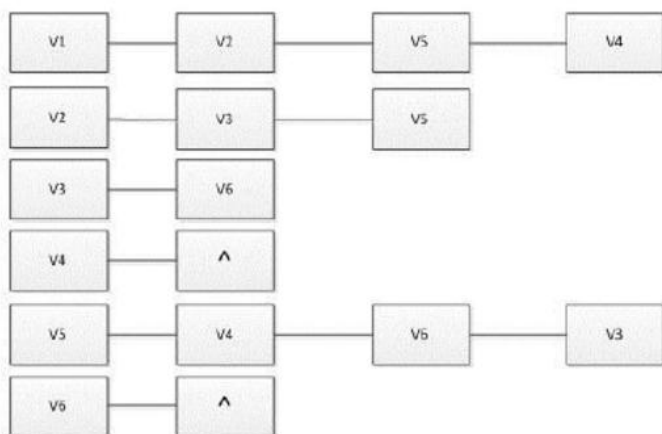
14、哈希查找中 k 个关键字具有同一哈希值，若用线性探测法将这 k 个关键字对应的记录存入哈希表中，至少要进行（ ）次探测。

[\(点击查看答案>>>>>>>>\)](#)

- A、k
- B、k+1
- C、 $k(k+1)/2$
- D、 $1+k(k+1)/2$

难度：★★★★★

15、G 是一个邻接表，以 V1 为起点，深度优先遍历序列 V1-V2-V3-V6-V5-V4，从 V1 开始的广度优先遍历序列是（ ）[\(点击查看答案>>>>>>>>\)](#)



- A、V1 V2 V5 V4 V3 V6
- B、V1 V2 V3 V4 V5 V6
- C、V1 V3 V5 V4 V2 V6
- D、V1 V2 V5 V3 V4 V6

难度：★★★★★



16、二叉树（ ）。在完全的二叉树中，若一个结点没有（ ），则它必定是叶结点。每棵树都能惟一地转换成与它对应的二叉树。由树转换成的二叉树里，一个结点 N 的左子女是 N 在原树里对应结点的（ ），而 N 的右子女是它在原树里对应结点的（ ）。(点击查看答案>>>>>>>>)

难度：★★

17、向量、栈和队列都是（ ）结构，可以在向量的（ ）位置插入和删除元素；对于栈只能在（ ）插入和删除元素；对于队列只能在（ ）插入和 5 删除元素。(点击查看答案>>>>>>>>)

难度：★★★

18、下列给定程序中，函数 fun 的功能是：把形参 a 所指数组中的最大值放在 a[0]中，接着求出 a 所指数组中的最小值放在 a[1]中，再把 a 所指数组元素中的次大值放在 a[2]中，把 a 数组元素中的次小值放在 a[3]中，依此类推。

例如，若 a 所指数组中的数据最初排列为：1、4、2、3、9、6、5、8、7，按规则移动后，数据排列为：9、1、8、2、7、3、6、4、5。形参 n 中存放 a 所指数组中数据的个数。

请在程序的下画线处填入正确的内容并将下画线删除，使程序得出正确的结果。

试题程序：

```
#define N 9
/*****found*****/
void fun(int a ① , int n)
{
    int i, j, max, min, px, pn, t;
    /*****found*****/
    for (i = 0; i < n - 1; i += ② )
    {
        max = min = a[i];
        px = pn = i;
        for (j = ③ ; j < n ; j ++ )
        {

            if (max < a[j])
            {
                max = a[j];
                px = j;
            }
            if (min > a[j])
            {
                min = a[j];
            }
        }
    }
}
```



```
pn = j;
}
}
if (px != i)
{
t = a[i]; a[i] = max; a[px] = t;
if (pn == i) pn = px;
}
if (pn != i + 1)
{
t = a[i + 1];
a[i + 1] = min;
a[pn] = t;
}
}
}
```

①_____

②_____

③_____

[\(点击查看答案>>>>>>>\)](#)

难度：★★★★★

19、线性表 (a_1, a_2, \dots, a_n) 的顺序存储结构中，设每个单元的长度为 L ，元素 a_i 的存储地址 $LOC(a_i)$ 为 () [\(点击查看答案>>>>>>>\)](#)

难度：★★★★★

20、有一个数组 $(53, 83, 18, 59, 38, 35)$ ，依次将其存储在 hash 表中，其中哈希函数为 $h(k)=k\%7$ ，如采用线性探测（每次向后查找 1 位）的方式解决冲突，则该 hash 表上查找 38, 35, 53 访问 hash 表的表项次数分别为 (), (), ()。 [\(点击查看答案>>>>>>>\)](#)

难度：★★★★★

[更多专项练习题目 >>>>>>>](#)



牛客网，国内最大的程序员学求职备考神器



3、扑克牌大小 [（点击实战练习>>>>>>>>>>）](#)

题目描述:

扑克牌游戏大家应该都比较熟悉了，一副牌由 54 张组成，含 3~A，2 各 4 张，小王 1 张，大王 1 张。牌面从小到大用如下字符和字符串表示（其中，小写 joker 表示小王，大写 JOKER 表示大王）：)

3 4 5 6 7 8 9 10 J Q K A 2 joker JOKER

输入两手牌，两手牌之间用“-”连接，每手牌的每张牌以空格分隔，“-”两边没有空格，如：4 4 4 4-joker JOKER

请比较两手牌大小，输出较大的牌，如果不存在比较关系则输出 ERROR

基本规则:

(1) 输入每手牌可能是个子, 对子, 顺子 (连续 5 张), 三个, 炸弹 (四个) 和对王中的一种, 不存在其他情况, 由输入保证两手牌都是合法的, 顺子已经从小到大排列;

(2) 除了炸弹和对王可以和所有牌比较之外，其他类型的牌只能跟相同类型的存在比较关系(如，对子跟对子比较，三个跟三个比较)，不考虑拆牌情况(如：将对子拆分成个子)

(3) 大小规则跟大家平时了解的常见规则相同，个子，对子，三个比较牌面大小；顺子比较最小牌大小；炸弹大于前面所有的牌，炸弹之间比较牌面大小；对王是最大的牌；

(4) 输入的两手牌不会出现相等的情况。

答案提示:

(1) 除了炸弹和对王之外，其他必须同类型比较。

(2) 输入已经保证合法性，不用检查输入是否是合法的牌。

(3) 输入的顺子已经经过从小到大排序, 因此不用再排序了.

输入描述:

输入两手牌，两手牌之间用“-”连接，每手牌的每张牌以空格分隔，“-”两边没有空格，如 4 4 4 4-joker JOKER。

输出描述:

输出两手牌中较大的那手，不含连接符，扑克牌顺序不变，仍以空格隔开；如不存在比较关系则输出 **ERROR**。



4、用两个栈实现队列 [\(点击实战练习>>>>>>>>>>\)](#)

题目描述:

用两个栈来实现一个队列，完成队列的 **Push** 和 **Pop** 操作。队列中的元素为 **int** 类型。

```
class Solution
{
public:
    void push(int node) {

    }

    int pop() {

    }

private:
    stack<int> stack1;
    stack<int> stack2;
};
```

5、第一个只出现一次的字符（[点击实战练习>>>>>>>>>](#)）

题目描述:

在一个字符串($1 \leq \text{字符串长度} \leq 10000$, 全部由字母组成)中找到第一个只出现一次的字符,并返回它的位置

```
class Solution {
public:
    int FirstNotRepeatingChar(string str) {

    }
};
```

更多经典编程练习 >>>>>>>



三、名企考试真题汇总

1、(华为)字符的集合（点击实战练习>>>>>>>>>>）

题目描述:

输入一个字符串，求出该字符串包含的字符集合

输入描述:

每组数据输入一个字符串，字符串最大长度为 100，且只包含字母，不可能为空串，区分大小写。

输出描述:

每组数据一行，按字符串原有的字符顺序，输出字符集合，即重复出现并靠后的字母不输出。

2、(美团)二维数组打印 (点击实战练习>>>>>>>>>>)

题目描述:

有一个二维数组 ($n \times n$)，写程序实现从右上角到左下角沿主对角线方向打印。
给定一个二位数组 `arr` 及题目中的参数 `n`，请返回结果数组。

```
class Printer {
public:
    vector<int> arrayPrint(vector<vector<int> > arr, int n) {
        // write code here
    }
};
```

3、(去哪儿)表达式合法判断(点击实战练习>>>>>>>>>>)

题目描述:

写一段代码,判断一个包括'{' , '[' , '(' , ')' , ']' , '}' 的表达式是否合法(注意看样例的合法规则。)

给定一个表达式 A, 请返回一个 bool 值, 代表它是否合法。

```
class ChkExpression {
public:
    bool chkLegal(string A) {
        // write code here
    }
};
```



题目描述:

路上，他经过了一个小镇，发现小镇的人们都围在一棵树下争吵。桂上前询问情况，得知小镇的人们正缺一个镇长，他们希望能选一个知名又公正的镇长，即，大家希望能选出一个人，所有人都认识他，但同时他不认识镇上除自己以外的其他人（在此，我们默认每个人自己认识自己）。可是小镇里的人太多了，一下子大家谁也说服不了谁。

现在你手上也拿到了这样一份认识关系的清单。其中上面给出的认识关系是单向的，即，A 认识 B 与 B 认识 A 是相互独立的，只给出 A 认识 B 就不能认为 B 认识 A，例如，我认识你，你不一定认识我。而且，这里的认识关系也不具有传递性，即，A 认识 B，B 认识 C，但这不代表 A 认识 C。同时，为了方便处理，这份清单中，镇上的 N 个人依次编号为 1 到 N。你能否像桂一样快速找到合适的镇长人选呢？

之后每组数据的第一行有 2 个整数 n 和 m ($1 \leq n \leq 105, 0 \leq m \leq 3 \times 105$), 依次表示镇上的人数和相互之间的认识关系数。

保证所有数据中 80%的数据满足 $n \leq 1000, m \leq 10000$

第一行，一个整数，表示你所找出来的合适的镇长人选人数 num i 。

特别的，如果并没有找到合适的镇长，第一行输出一个数 0，第二行留空即参见样例)。



5、(华为)字符集合[（点击实战练习>>>>>>>>>>）](#)

题目描述:

输入一个字符串，求出该字符串包含的字符集合

输入描述:

每组数据输入一个字符串，字符串最大长度为 100，且只包含字母，不可能为空串，区分大小写。

输出描述:

每组数据一行，按字符串原有的字符顺序，输出字符集合，即重复出现并靠后的字母不输出。

更多名企真题练习 >>>>>>>