

## 一、填空题

1 逻辑结构、存储结构、算法

2 链式存储结构

3  $O(1)$

4  $8 \ 4+4+3+4$  条边，一共 16 个结点，叶节点为 8，也可以画图试试

5 BFGDECA 先找根节点到 A，A 的左子树 B，右子树 FDGCE，右子树的根节点 C，C 的左子树 FDG,右子树 E，左子树的根节点 D，D 的左子树 F，右子树 G

6  $n-m+1$ ，对于任何一个有子节点的结点而言，其子节点有且只有一个子节点右子树为空，再加上根结点也是没有右子树的

7 子图

8 4 次 依次比较 65, 15, 30, 37

9  $n(n-1)/2 \ 0+1+\dots+n-1$

10 1, 5, 11, 15, 19, 77, 59, 48, 26, 61

## 二、简答题

1. 矩阵总元素数量是  $100 \times 100 = 10000$ ，非零元素数量为 200， $200/10000 = 0.02 < 5\%$ ，所以是稀疏矩阵

2. 书上的概念，没啥好说的

3. 起泡排序法，因为选择排序每趟排序只会交换两个元素，第二趟交换了三个元素的位置，而泡排序每趟排序将最大值像泡一样交换到数组最右边，同时每趟排序过程中，如果相邻元素不符合要求的顺序，会交换两个相邻元素，符合所给结果

4 最大递归深度是  $n$ ，最小递归深度是  $\log_2(n+1)$ 取上界

### 三、综合题

1. 第四句,  $p \rightarrow rlink \rightarrow llink = p$  做过类似的题目, 修改四个指针, 前两句做的是修改  $p$  的左右指针, 没什么问题, 后两句做的是修改  $q$  的右指针和原  $q$  的右结点的左指针, 但第四句错了, 因为  $q$  的右结点已经修改了

2. 又是一道似曾相识的题目, 有两种大的情况, 第 7 层只有叶节点, 还有一种情况是第 7 层还有分支结点。  $10+2^6-1=73$ ,  $(2^6=64, 64-10=54, 2^7-1=127, 127+108(107)=235(234))$

3. 从度的角度更好一点, 将顶点编号为  $1, 2, \dots, n$ , 第 1 个顶点可以跟其他顶点有  $n$  条边, 除去顶点 1, 第 2 个顶点可以跟其他顶点有  $n-1$  条边, 而第  $n-1$  个顶点可以跟第  $n$  个顶点有一条边, 归纳总结为  $n+(n-1)+\dots+1=(n-1) \times n/2$

4. 90, 30, 50, 10, 80, 20 || 90, 80, 50, 10, 30, 20 || 90, 80, 50, 10, 30, 20 || 90, 80, 50, 30, 10, 20 || 90, 80, 50, 30, 20, 10

### 四、算法设计题

判断一个序列是否是有向图的拓扑序列, 简单

`typedef int vtype;`//防止无法编译用, 实际做题删除

`typedef struct edge`

`{`

`int adjvex;`

`struct edge*next;`

`}ELink;`

```
typedef struct ver
```

```
{
```

```
    int indegree;
```

```
    vertype vertex;
```

```
    ELink*link;
```

```
}VLink;
```

```
int isTopologicalSequence(VLink G[],int V[],int n)
```

```
{
```

```
    //按照序列，挨个检查每个结点是否可以作为拓扑序列中的点
```

```
    int i;
```

```
    int indegree[n];
```

```
    for(i = 0; i < n; i++)
```

```
        indegree[i] = G[i].indegree;
```

```
    int current = V[0];
```

```
    for(i = 1; i < n && indegree[current] == 0; i++)
```

```
    {
```

```
        //将 current 所连接的结点入点减一
```

```
        ELink*p = G[current].link;
```

```
        while(p)
```

```
        {
```

```
            indegree[p->adjvex]--;
```

```

        p = p->next;

    }

    //接下来判断下一个序号能否作为拓扑序列

    current = V[i];

}

if(!indegree[current])

    return 1;

return 0;

}

```

## 五、单项选择题

1. 选 C
2. 选 D
3. 选 A，从简单的选项开始看 BC 明显都是无限循环，再看 D，没用 break 也没用循环结束条件，所以考试的时候直接选 A，再来看看 A，循环条件是  $a > ++b$ ，而刚进循环就不满足条件
4. 选 C 显然这是数组指针，ptr 是一个指向大小为 N 一维数组的指针
5. 选 B，选项 A 中后一句 r 没有分配空间，也不能将字符串赋值给字符，选项 B 可以，但换成数组就不行，所以 C 错了，D 中 temp 没有声明，即使声明了也不能这么做
6. 选 D，很显然只包括一个函数头，所以是对函数的声明，但没有这个答案，那显然是换了一个说法，显然 BC 是错的，A 对了吗？思考一下定义和声明的区别，这条语句后面加了一个分号，所以不是定义，那选 D，D 的说法也可以接受

7. 选 A，考的是一个基本的字符串输入输出问题，pq 共享同一段空间，后输入的字符串会覆盖前面的字符串

8. 选 D，送分题

9. 选 B，做字符串替换  $a = ++k + 1 * k + 1$ ， $a = 9$

10. 选 A，送分题

## 六、简答题

1. 头文件本质上是已经写好的 C 程序，可以降低程序开发的难度，与源程序同时编译

2. 这个讲过

3. 书上好像也有，(1) 存储区域，全局变量存储在数据区，局部变量存储在堆栈中 (2)

作用范围 (3) 生命周期

4. 相同，都是指针

## 七、填空题

1. 50 x

2.  $-1 * 2^i + 1$

3.  $c = c + 5$   $c = c - 21$

4.  $*t * s - *t$

5.  $str3[k] = str2[j++]$   $str1[i] == '\0'$ ，记住是单引号

6.  $argc > 1$   $*argv$

7. a r fp2 fgetc(fp2)

8. 求整数 n 的位数，答案给的是正整数

9. 从上往下分析，n 是 str 的实际长度，ptr1 指向 str 的首地址，ptr2 指向 str 的尾地址（实际上是最后一个字符的地址），while 循环比较的是地址，后面做的是比较 ptr1 和 ptr2 中存储的字符，如果 ptr1 中的字符和 ptr2 中的字符相同，则继续循环否则跳出循环，最后比较两者的地址，实际上循环做的是挨个比较 ptr1 和 ptr2 的字符，ptr1 不断递加，ptr2 递减所以是比较一个字符串从前到后和从后到前的字符，回文字符串

10. 以追加“a”方式向文件写入“data”，输出此时文件指针的位置

## 八、程序设计题

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    char str[100];

    char c;

    gets(str);

    c = getchar();

    int index = 0;

    while(str[index] && str[index] != c)

        index++;

    if(!str[index])

        printf("未找到字符%c", c);

    else

    {
```

```
printf("%d\n",index);

printf("原字符串:%s\n",str);

while(str[++index])

{

    str[index-1] = str[index];

}

str[index-1] = '\0';

printf("删除后字符串:%s\n",str);

}


return 0;

}
```