

2017 年北京航空航天大学招收硕士研究生

入学考试试题参考答案

一、解:

| 题号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 答案 | D | B | A | B | A | C | C | D | D | B |

二、解:

1. 顺序

2. 链式

3. 2

4. $n_0 = (k-1) * n + 1$

5. 中序遍历

6. $2(n-1)$

7. 队列结构

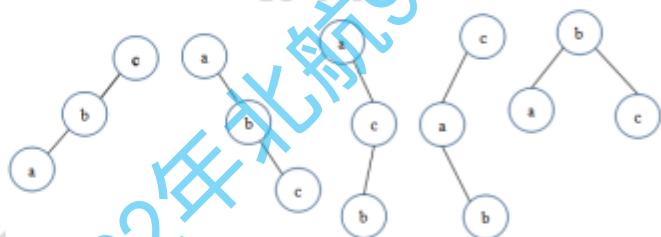
8. 哈希(散列)查找法

9. 12 次

10. $O(n^2)$, $O(n^2)$, $O(n \log_2 n)$

三、解:

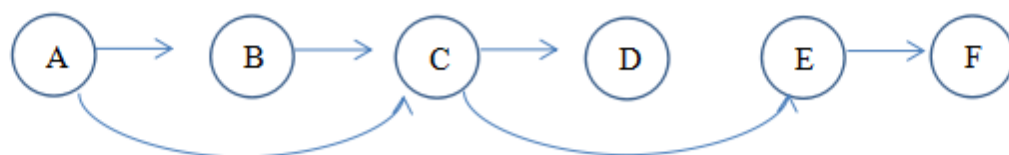
1.



①. 还原为邻接矩阵表示形式

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | - | 1 | 1 | 0 | 0 | 0 |
| B | | - | 1 | 0 | 0 | 0 |
| C | | | - | 1 | 1 | 0 |
| D | | | | - | 0 | 1 |
| E | | | | | - | 1 |
| F | | | | | | - |

②. 还原成有向图



③. 写出可能的拓扑序列：ABCDEF 或者 ABCEDF

3. 算法各自有各自的适用场合。根据数据原始状态不同，排序过程性以及结果稳定性要求不同，采用不同的排序算法。

4. 第一趟: 49 13 27 50 76 38 65 97

第二趟: 27 13 49 38 65 50 76 97

第三趟: 13 27 38 49 50 65 76 97

四、算法核心思想：算法的基本思想比较简单，即从链表的第 2 个结点（用指针变量 p 指向该结点）开始，依次判断在该结点之前是否出现过其数据域值相同的结点，若出现过，则删除 p 指向的结点；否则，将 p 的指向后移一个结点，重复上述过程，直到链表中所有结点都处理完毕（此时有 $p == \text{NULL}$ ）

```
void DEL3(LinkList list)
```

```
{
```

```
    Linklist p, q, r, flag=0;
```

```
    p=list->link; /*从链表的第2个结点开始寻找多余结点*/
```

```
    r=list;
```

```
    while(p!=NULL){
```

```
        q=list; /*每次从链表的第一个结点开始寻找多余结点*/
```

```
        while(q!=p)
```

```
            if(q->data==p->data){ /*p所指结点为多余结点*/
```

```
                r->link=p->link; /*删除p所指结点*/
```

```
                free(p);
```

```
                flag=1;
```

```
                break;
```

```
            }
```

```
        else
```

```
            q=q->link;
```

```

        if(flag){
            p=r->link;
            flag=0;
        }
        else{
            r=p;
            p=p->link;      /*准备判断下一个结点是否为多余结点*/
        }
    }
}

```

五、解：

| 题号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|----|
| 答案 | D | C | B | B | C | A | C | A | B | D |

六、解：

1.答：switch(mo)

```

{
    case1:case3:case5:max=31;break;
    case2:max=28;break;
    case4:case6:max=30;break;
    default :max=0;
}

```

2.答：P1,P2,P3 大小相同，因为不论指向什么类型，指针类型都用于存放地址。对 P1 来说其指存的地址指向 int 类型所占字节（2 个或者 4 个）的第一字节，P2 也是同理，P3 中的地址则指向函数入口地址，所以它们所占大小都是一样的。

3.答：①声明的关键字不同。结构体 struct，共用 union。

②所占存储空间大小不同：结构体所占空间是所有成员变量所占空间之和，而共用体则是成员变量中所占空间最大变量所占的字节数。

③共用体变量中实际只放一个值，新赋值的变量会覆盖旧变量，而且所有变量存放的起始地址都一样；结构体变量中的各成员变量则可以被分别赋值。

4.答：

```
void main(){
    FILE *fp;
    char *str1,*str2;
    if((fp=fopen("c://tag.txt","r"))==NULL) {
        printf("the file is empty");
        return;
    }
    if(fgetc(str1,100, fp)==NULL) {
        printf("the file is empty");
        return;
    }
    str2=str1;
    while(fgets(str1, 100, fp) !=NULL)
        str2=str1;
    puts(str2);
    fclose(fp);
}
```

七、程序设计题（本题共 15 分）

```
float mul(int a,int n) {
    float mul_sum=1;
    int i;
    for (int i=0;i<a;++i)
        mul_sum*=n;
    return mul_sum;
}

float psum(int a,int n) {
    float sum=0;
    int i;
    if(a<=0 || n<=0)
        return sum;
    for (int i=1;i<n;++i)
        sum+=mul(a,i);
    return sum;
}
```

```

}
```

八. 程序设计题（本题共 20 分）

```

#include<stdio.h>
```

```

#define M 80
```

```

int stoi(char*p,int n)//将数字字符数组变为整型
```

```

{
```

```
    int a=0,i;
```

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

```
        a=a*10+(*p++-'0');
```

```
    return a;
```

```

}
```

```

void output(int a[],int n)//选择排序输出数组
```

```

{
```

```
    int i,j,max,temp;
```

```
    for(i=0;i<n-1;i++)
```

```
    {
```

```
        max=i;
```

```
        for(j=i+1;j<n-1;j++)
```

```
        {
```

```
            if(a[max]<a[j])
```

```
                max=j;
```

```
        }
```

```
        if(max!=i)
```

```
        {
```

```
            temp=a[i];
```

```
            a[i]=a[max];
```

```
            a[max]=temp;
```

```
        }
```

```
        printf("%d ",a[max]);
```

```
    }
```

```

}
```

```

void main()
```

```

{
```

```
char str[M],s[M];//str 用来存放输入的字符串，s 用来存放提取出的数字字符串
int a[M];//存放整型数组
int i,j,k,l=0;
printf(“请输入一行字符串\n”);
gets(str)
for(i=0;str[i]!='\0';i++)
{
    k=0;
    for(j=i;str[j]!='\0'&&str[j]>=48&&str[j]<=57;j++)
        s[k++] =str[j];
    s[k]='\0';
    a[l++]=stoi(s,k);
}
output(a, l);
}
```