

机密★启用前

西南交通大学 2018 年硕士研究生
招生入学考试试卷

试题代码: 840

试题名称: 数据结构与程序设计

考试时间: 2017 年 12 月

考生请注意:

1. 本试题共四题, 共 10 页, 满分 150 分, 请认真检查;
2. 答题时, 请直接将答题内容写在考场提供的答题纸上, 答在试卷上的内容无效;
3. 请在答题纸上按要求填写试题代码和试题名称;
4. 试卷不得拆开, 否则遗失后果自负。

一、单项选择题 (52 分, 共 26 小题, 每小题 2 分) (答在试卷上的内容无效)

1. 在数据结构中, 与所使用的计算机无关的是数据的_____结构。
A. 逻辑 B. 存储 C. 逻辑和存储 D. 物理
2. 循环单链表的头结点指针为 head, 尾结点指针为 p, 若该链表中至少有 2 个结点, 则以下关系运算表达式运算结果为“真”的是_____。
A. $p \rightarrow next == NULL$ B. $p \rightarrow next == head$
C. $p == NULL$ D. $p == head$
3. 长度为 n 的顺序表删除下标为 i 的元素(已知元素下标范围为 $0..n-1$), 需要向前移动_____个元素。
A. $n-i$ B. $n-i+1$ C. $n-i-1$ D. $i+1$
4. 将长度为 n 的单链表连接在长度为 m 的单链表之后的算法的时间复杂度为_____。(算法假设两个单链表仅已知头结点指针)
A. $O(1)$ B. $O(n)$ C. $O(m)$ D. $O(m+n)$
5. 五节车厢以编号 1, 2, 3, 4, 5 顺序进入铁路调度站(栈), 可以得到列车编组是_____。
A. 3, 4, 5, 1, 2 B. 3, 5, 4, 2, 1
C. 1, 3, 5, 2, 4 D. 2, 4, 1, 3, 5
6. 设有两个串 S1 和 S2, 求串 S2 在 S1 中首次出现位置的运算称作_____。
A. 连接 B. 提取子串 C. 比较大小 D. 模式匹配(子串查找)

7. 在一棵度为 3 的树中, 度为 3 的结点个数为 1, 度为 2 的结点个数为 2, 则度为 0 的结点个数为_____。
- A. 4 B. 5 C. 6 D. 7
8. 下列关于有向无环图(DAG)的拓扑排序序列的叙述中, 正确的是_____。
- A. 存在且惟一 B. 存在且不惟一
C. 存在但可能不惟一 D. 可能不存在
9. 无向图的邻接矩阵是_____。
- A. 对称矩阵 B. 稀疏矩阵 C. 上三角矩阵 D. 对角矩阵
10. 若图 G 的邻接表中有奇数个表结点, 下列选项中正确的是_____。
- A. G 中必有奇数个顶点 B. G 中必有偶数个顶点
C. G 为无向图 D. G 为有向图
11. 16 个元素的有序表进行对分(折半)查找, 成功查找的最大关键比较次数是_____。
- A. 3 B. 4 C. 5 D. 6
12. 二路归并排序的平均时间复杂度为_____。
- A. $O(n\log_2 n)$ B. $O(n^2)$ C. $O(n)$ D. $O(\log_2 n)$
13. 一棵哈夫曼树总共有 23 个结点, 该树共有_____个叶结点(终端结点)
- A. 11 B. 12 C. 13 D. 14
14. 在 C 语言中, int 型数据在内存中的存储形式是_____。
- A. 补码 B. 反码 C. 原码 D. ASCII 码
15. 下述语句执行后, 变量 k 的值是_____。
- ```
int k=1;
while (k++<10);
```
- A. 10    B. 11    C. 9    D. 无限循环
16. C 语言函数 strcmp(s,t)的功能是\_\_\_\_\_。
- A. 比较字符串 s 和 t 的长度    B. 比较字符串 s 和 t 的大小  
C. 比较字符串 s 和 t 是否相等    D. 将字符串 t 赋给字符串 s
17. 已知数组 `int a[5]={10,20,30,40,50}`, 指针 `int *p=a+1`; 则表达式 `*p++` 的值是\_\_\_\_\_。
- A. 20    B. 30    C. 21    D. 31
18. 若 fp 是指向某文件的指针, 若在文件最后一个字节之后进行了读操作, 则库函数 feof(fp)的返回值是\_\_\_\_\_。
- A. -1    B. 0    C. NULL    D. 非零值
19. 若已定义 x 为 int 类型变量, 下列语句中说明指针变量 p 的正确语句是\_\_\_\_\_。



A. int p=&x;    B. int \*p=x;

C. int \*p=&x;    D. \*p=\*x;

20. 以下程序的输出结果是\_\_\_\_\_。

```
#define M(x,y,z) (x)*y+z
void main()
{ int a,b,c,
 a=b=1,c=3,
 printf("%d\n", M(a+b,b+c,c+a));
}
```

A. 9    B. 12    C. 8    D. 其它

21. 下列选项中正确的语句组是\_\_\_\_\_。

A. char s[10]; s={"China"};    B. char \*s; s={" China"};

C. char s[10]; s=" China ";    D. char \*s; s=" China";

22. 下面的程序执行后, 文件 test.txt 中的内容是\_\_\_\_\_。

```
#include <stdio.h>
void fun(char *fname, char *st)
{ FILE *myf;
 int i;
 myf=fopen(fname, "w");
 for(i=0; i<strlen(st); i++)
 fputc(st[i], myf);
 fclose(myf);
}
void main()
{
 fun("test.txt", "Welcome to ");
 fun("test.txt", "Chengdu");
}
```

A. Chengdu    B. Welcome to Chengdu

C. Welcome to    D. to

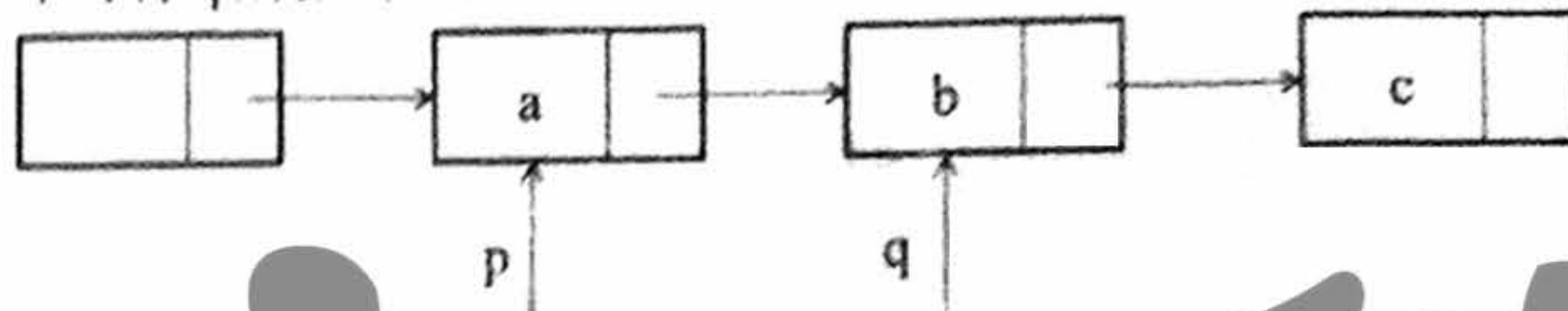
23. 运行下面程序, 则输出结果是\_\_\_\_\_。

```
#include <string.h>
void main()
{ char *p="abcde\0fghjik\0";
 printf("%d\n", strlen(p));
}
```

}

A. 12      B. 15      C. 6      D. 5

24. 若已建立以下的链表结构, 指针  $p$ ,  $q$  分别指向图中所示结点, 则以下选项中可将  $q$  所指的结点从链表中删除并释放该结点的语句是\_\_\_\_\_。

A.  $(*p).next = (*q).next; free(p);$ B.  $p = q->next; free(q);$ C.  $p = q; free(q);$ D.  $p->next = (*q).next; free(q);$ 

25. 下面程序的运行结果是\_\_\_\_\_。

```
#include <stdio.h>
```

```
void main()
```

```
{ int y=10;
```

```
do{ y--; } while(--y);
```

```
printf("%d\n", y--);
```

```
}
```

A. -1      B. 1      C. 0      D. 8

26. 下列程序执行后的输出结果是\_\_\_\_\_。

```
#include <stdio.h>
```

```
void main()
```

```
{ int x='d';
```

```
printf("%c\n", 'A'+(x-'a'+2));
```

```
}
```

A. F      B. H      C. J      D. E

二、填空题 (30 分, 共 30 空, 每空 1 分) (答在试卷上的内容无效)

请在答题纸上按填空号(1), (2), (3), ..., 顺序写出各空答案!

◆ 数据元素之间的关系在计算机中有两种不同的表示方法: 顺序映象和非顺序映象, 并由此得到两种不同的存储结构: 顺序存储结构和 (1) 存储结构。

◆ 子串 "ab" 在主串 "aababaaba" 中的匹配次数是 (2) 。

◆ 完全二叉树共有 1024 个结点, 则该完全二叉树的高度是 (3) , 叶子结点数是 (4) 。

◆ 直接插入排序的平均时间复杂度为  $O(n^2)$ , 当待排序的初始关键字序列正序时, 直接插入排序的时间复杂度减低为  $O$ ( (5) ) 。



◆ 分块索引顺序表查找关键字时, 假设索引表和分块内部均采用顺序查找, 若查找表中共有 100 个关键字, 则使得平均查找长度最优的最佳分块数为 (6)。

◆ 拥有 5 个结点的二叉树先序和中序遍历结点访问次序均为 ABCDE, 则其后序遍历结点访问次序为 (7)。

◆ 设有一个 10 行×10 列的对称矩阵 A, 采用压缩存储的方式, 将其下三角部分(含主对角线)以行序为主存储到一维数组 B 中(数组及矩阵下标均从 1 开始), 则矩阵中 8 行 5 列元素在一维数组 B 中的下标是 (8)。

◆ 已知带附加头结点单向链表结点及其指针数据类型定义如下:

```
typedef struct node
{ int data; struct node *next;
} LNode, *LPtr;
```

以下算法在各数据结点按 data 域值由小到大次序连接的升序链表中插入一个新结点, 使链表插入新结点后各结点连接次序保持升序, 请填空。

```
void insertUpSortedLinList(LPtr h, Lptr s) //h, s 分别指针附加头结点和新结点
{ LPtr pr=h, p= (9);
 while(p&& (10)) //搜索插入位置, 结点*s 应插入到*pr 和*p 之间
 { pr=p; p=p->next;
 }
 pr->next= (11); s->next= (12); //插入结点*s
}
```

◆ 已知二叉树的二叉链表结点数据类型定义如下:

```
typedef struct node
{ int bf; struct node *lchild, *rchild;
} BTreeNode, *BT;
```

以上结点结构中, bf 表示结点的平衡因子(定义为该结点的左子树与右子树的高度差)。以下算法实现所有结点 bf 域的正确填充, 请填空。

```
int fill_bf(BT bt) //bt 为根结点地址, 函数返回值为该二叉树的高度
{ if(!bt) return 0; //空树高度为 0
 int hL=fill_bf(bt->lchild); //求左子树高度
 int hR=fill_bf((13)); //求右子树高度
 bt->bf= (14) ; //访问根结点: 填充平衡因子
 int h=hL>hR?hL:hR;
 return (15) ; //返回树高(用含 h 的表达式表示)
}
```

- ◆ 在 C 语言中的实型变量分为两种类型, 它们是 float 和 (16)。

- ◆ 定义 `int a = 12, b = -1;` 执行完下面语句后, `a = (17)`, `b = (18)`。

```
if (a++ > 0 || ++b == 0)
```

```
{
```

```
 b *= 10;
```

```
}
```

- ◆ 当 `a=5, b=4, c=2` 时, 表达式 `a>b!=c` 的值是 (19)。

- ◆ 若一个派生结构的定义为

```
struct People
```

```
{ char name[20];
```

```
 float score;
```

```
 union {int key; char keys[20];} id ;
```

```
};
```

则表达式 `sizeof(struct People)` 的值为 (20)。

- ◆ 已知 D 盘根目录上存在一个文本文件 `a.txt`, 如果仅仅需要读取该文件信息, 并以字符文件进行操作, 则打开语句为: `FILE *fp = (21)`。

- ◆ 已知二维数组 `int a[][4]={1,2,3,4,5,6,7,8,9,10,11,12};` 定义向量指针 `int (*p)[4]=a;`

则 `(++p)[0][1] = (22)`

- ◆ 在 C 语言中, 表达式为逻辑值“假”时用 (23) 表示。

- ◆ 设 `x, y, z` 和 `k` 都是 `int` 型变量, 则执行表达式: `x=(y=4, z=16, k=32)` 后, `x` 的值为 (24)。

- ◆ 有如下程序段, 输入数据: `12345ffl678` 后, `u` 的值是 (25), `v` 的值是 (26)。

```
int u;
```

```
float v;
```

```
scanf("%3d%f", &u, &v);
```

- ◆ 下列程序段中循环体被执行的次数是 (27)。

```
n=1234;
```

```
do n/=10;
```

```
while (n);
```

- ◆ 有以下程序, 为使之正确运行, 应增加的 `#include` 指令是 (28)。

```
#include <stdio.h>
```

```
Void main()
```

```
{ double x=2.5, y=3.9;
```

```
 printf ("%lf\n", fabs(y/x));
```

```
}
```

- ◆ 设 `x` 和 `y` 均为 `int` 型变量, 则以下 `for` 循环中的 `scanf` 语句最多可执行的次数是 (29)。

```
for (x=0, y=0; y!=123&& x<3; x++)
```

```
 scanf ("%d", &y);
```



◆ 已知  $a=13, b=6$ ,  $a \& b$  的十进制数值为 (30)。

三、简答题 (48 分, 共 10 小题, 前 4 小题每题 6 分, 后 6 小题每题 4 分) (答在试卷上的内容无效)

1. 已知某稀疏矩阵  $A$  的三元组顺序表如下所示 (表中  $a_{ij}$  表示  $A$  矩阵的  $i$  行  $j$  列元素), 试写出该矩阵的转置矩阵的三元组顺序表。(6 分)

| $i$ | $j$ | $a_{ij}$ |
|-----|-----|----------|
| 1   | 3   | -5       |
| 2   | 4   | 6        |
| 2   | 5   | 3        |
| 4   | 2   | -8       |
| 5   | 2   | 9        |
| 5   | 5   | -4       |

2. 十个关键字组成的初始序列 54, 76, 28, 39, 87, 46, 25, 78, 62, 12 进行由小到大排序, 请回答以下问题。(此题 6 分)

(1) 以第一个元素 54 作为支点进行快速排序, 写出一趟划分得到的序列。

(2) 进行增量为 3 的一趟希尔排序, 写出得到的序列。

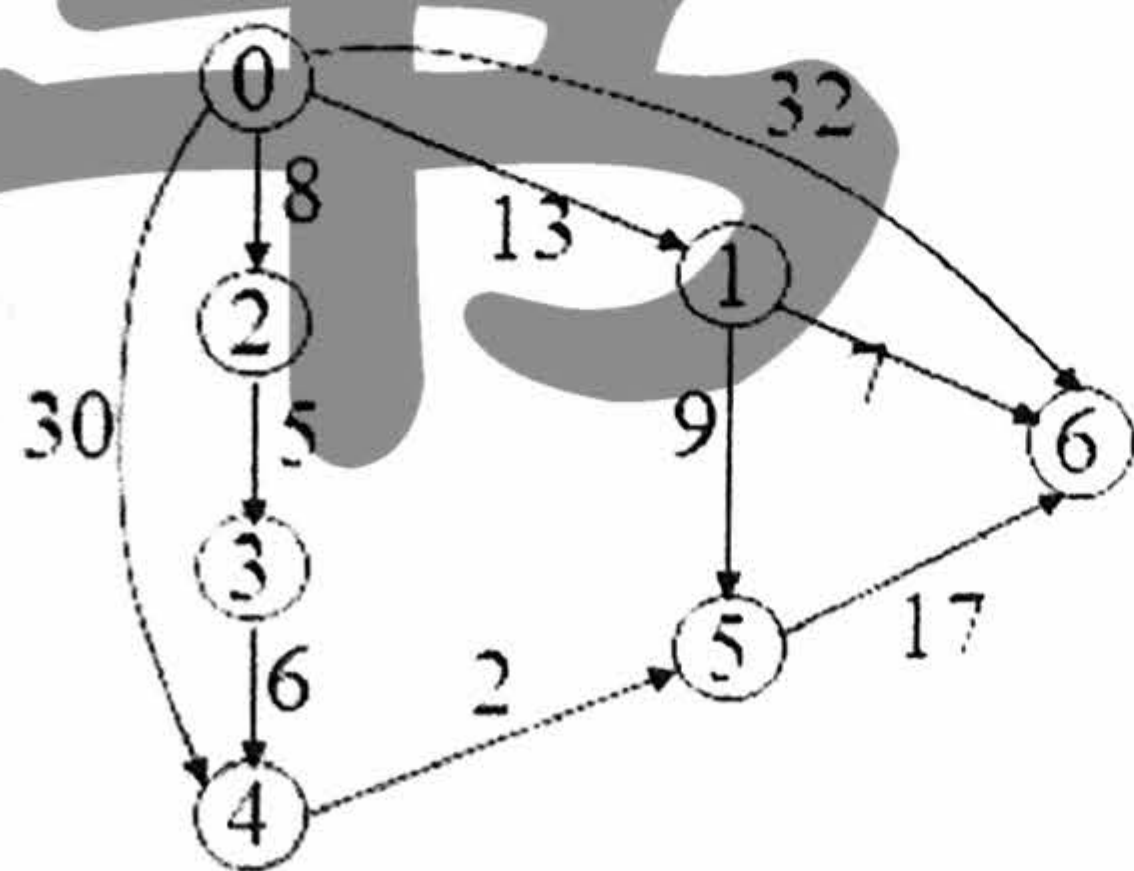
(3) 进行堆排序, 写出初始建大根堆后得到的序列。

3. 依次把 8 个关键字 80, 50, 20, 90, 30, 70, 60, 40 插入到初始状态为空的二叉排序树。(此题 6 分)

(1) 画出最终得到的含有 8 个关键字的二叉排序树;

(2) 8 个关键字查找概率相同, 计算成功查找时的平均查找长度  $ASL_{成功}$ 。

4. 写出以下有向网中, 0 号顶点至其它顶点的最短带权路径长度。(6 分)



5. 写出下面程序的运行结果。(4 分)

```
#include <stdio.h>
void main()
{ FILE *fp;
```

```
int i=10, j=30, k, n;
fp=fopen("dl.dat", "w");
fprintf(fp, "%d\n", i);
fprintf(fp, "%d\n", j);
fclose(fp);
fp=fopen("dl.dat", "r");
fscanf(fp, "%d%d", &k, &n);
printf("%d, %d\n", k, n);
fclose(fp);
}
```

6. 写出下列程序段的输出。(4 分)

```
void main()
{ char str[] = "aBcDeFgH", *p;
 for(p=str; *p; p++)
 if(*p>=65&&*p<=90) putchar(*p-65);
}
```

7. 写出以下程序的输出结果。(4 分)

```
#include <stdio.h>
void main()
{ int i=0, j=0, k=0, m;
 for (m=0; m<4; m++)
 switch (m)
 { case 0: i=m++;
 case 1: j=m++;
 case 2: k=m++;
 case 3: m++;
 }
 printf("\n%d, %d, %d, %d", i, j, k, m);
}
```

8. 写出下面程序的输出结果。(4 分)

```
void swap(char *x, char *y)
{ char t;
 t=*x; *x=*y; *y=t;
}

void main() {
 char *s1="abc", *s2="123";
 swap(s1, s2);
}
```



```
 printf("%s, %s\n",s1, s2);
 }
```

9. 写出以下程序的输出结果。(4 分)

```
#include <stdio.h>

void f(int i)
{ static int j=0;
 j++;i++;
 printf("%d%d",i,j);
}

void main()
{ int m;
 for(m=0; m<4; m++) f(m);
}
```

10. 写出以下程序的运行结果。(4 分)

```
#include "stdio.h"
#include "stdlib.h"
typedef struct st
{ int d; struct st *next;
}ST;

void main()
{
 ST *head = (ST *)malloc(sizeof(ST));
 int a[10] = {4,5,6,7,8,44,55,66,77,88};
 int i; ST *p;
 for (i = 0; i < 10; i++){
 ST *node = (ST *)malloc(sizeof(ST));
 node->d = a[i];
 node->next = head->next;
 head->next = node;
 }
 p = head->next;
 for (i = 0; i < 10; i++){
```

```
 printf("%d",p->d); p=p->next;
 }
}
```

四、算法及程序设计题 (20 分, 共 2 小题, 每小题 10 分) (答在试卷上的内容无效)

1. 算法设计:

二叉树结点指针类型定义为

```
typedef struct node { int data; struct node *lchild, *rchild; } BT;
```

编写一个算法函数, 形参传入二叉树根结点地址 root, 函数值返回二叉树所有结点的 data 域值之和。

2. 程序设计:

要求计算任意一天在本年中为第几天 (每年的 1 月 1 日为该年的第 1 天)? (注意考虑闰年问题: 能被 4 整除但不能被 100 整除, 或能被 400 整除的年份即为闰年)

(1) 要求写一个函数 Days, 实现上面的计算。

(2) 定义一个日期结构体 DAY (包括年、月、日), 保存日期信息

(3) 要求日期由用户输入, 将其保存为 DAY 结构体的一个变量, 由主函数将该变量传递给 Days 函数

(4) 计算结果在主函数中进行输出。