

北京工业大学 2016 年硕士研究生入学考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

★所有答案必须做在答题纸上, 做在试题纸上无效

第一部分 数据结构

一、单项选择题 (本大题共 10 小题, 每小题 2 分, 共 20 分) 在每小题列出的四个选项中只有一个选项是符合题目要求的, 请将正确选项前的字母填在题后的括号内。

1. 设一棵二叉树共有 20 个度为 2 的结点, 则叶子结点共有 () 个。
A. 40 B. 19 C. 20 D. 21
2. 在顺序表中, 逻辑上相邻的元素在物理位置上 ()。
A. 相邻 B. 不相邻 C. 不一定相邻 D. 不确定
3. 不属于链表特点的是 ()。
A. 适用于频繁插入 B. 适用于频繁删除 C. 存取速度快 D. 方便扩充
4. 设有一个顺序栈 S, 元素 $S_1, S_2, S_3, S_4, S_5, S_6$ 依次进栈, 如果 6 个元素的出栈顺序为 $S_2, S_3, S_4, S_6, S_5, S_1$, 则顺序栈的容量至少为 ()。
A. 1 B. 2 C. 3 D. 4
5. 假设利用一个一维数组 `elements[maxSize]` 作为循环队列元素的存储结构, 并且通过指针 `front` 指向队头位置, 指针 `rear` 指向实际队尾位置的后一位置, 那么循环队列满的条件是 ()。
A. `front == rear` B. `rear == maxSize-1`
C. `rear == maxSize` D. `(rear+1)%maxSize == front`
6. 若按 () 列出二叉搜索树中所存储的元素, 则恰好是集合中所有元素从小到大的排序。
A. 先序 B. 中序 C. 后序 D. 按层次
7. 以下排序算法中, () 排序在一趟结束后不一定能选出一个元素放在其最终位置上。
A. 直接选择 B. 归并 C. 冒泡 D. 堆
8. 对于含有 n 个顶点 e 条边的连通图, 利用 Prim 算法求最小生成树的时间复杂度为 ()。
A. $O(n^2)$ B. $O(e^2)$ C. $O(n \log_2 n)$ D. $O(e \log_2 e)$
9. 堆的形状是一棵 ()。
A. 二叉排序树 B. 满二叉树 C. 完全二叉树 D. 平衡二叉树
10. 将 18 个元素的有序表存放在一维数组 `A[18]` 中, 第一个元素存放于 `A[0]` 中, 现进行二分查找, 则查找 `A[2]` 的比较序列的下标依次为 ()。

北京工业大学 2016 年硕士研究生入学考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

A. 0, 1, 2 B. 8, 4, 1, 2 C. 8, 4, 2 D. 8, 3, 1, 2

二、简答题（本大题共 4 小题，每小题 6 分，共 24 分）

1. 顺序表的插入和删除要求仍然保持各个元素原来的次序。设在等概率的情况下，对有 127 个元素的顺序表进行插入，平均需要移动多少个元素？删除一个元素，平均需要移动多少个元素？
2. 用邻接表表示图时，顶点个数设为 n ，边的条数设为 e ，在邻接表上执行有关图的遍历操作时，时间代价是 $O(n \cdot e)$ ？还是 $O(n+e)$ ？或者是 $O(\max(n,e))$ ？为什么？
3. 假设待排序序列的初始排列是正序的，则直接插入排序、直接选择和冒泡排序哪一个更好？反序呢？为什么？
4. 回答下列问题：

(1) 直接在二叉搜索树中搜索关键码为 key 的对象与从中序遍历输出的有序序列中顺序搜索关键码为 key 的对象，其效率是否相同？

(2) 输入关键码有序序列来构造一棵二叉搜索树，然后对此树进行搜索，试分析其效率。

三、（本题 12 分）请回答有关二次散列的问题：

用二次散列解决冲突，求下一个“空位”的探查序列为：

$$H_i = (H_0 + i^2) \% m, \quad H_i = (H_0 - i^2) \% m, \quad i = 1, 2, \dots, m/2.$$

其中， H_0 是第一次求得的散列地址， H_i 是第 i 次求得的散列地址， m 是散列表的大小。

- (1) 相邻的地址 H_i 与 H_{i-1} 之间是什么关系？
- (2) 为保证散列地址序列的地址不会循而往复地重迭， m 应设为什么数？
- (3) 为保证在删除时不中断查找链，可对被删记录做逻辑删除。为此，每个散列表的表项有 3 个状态，除了 Active（正在使用）和 Deleted（删除）状态外，还应有一个什么状态？

四、（本题 20 分）已知图 1 (a) 与图 1 (b) 均为 3 阶 B 树：

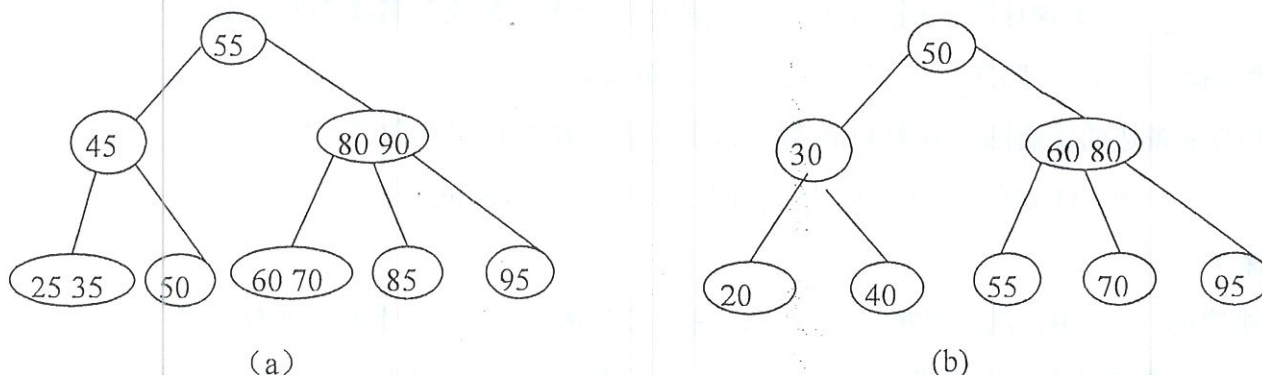


图 1

北京工业大学 2016 年硕士研究生入学考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

(1) 试分别画出在图 1 (a) 中插入 65, 15, 40, 30 之后 B 树的变化;

(2) 试分别画出在图 1 (b) 中删除 50, 40 之后 B 树的变化。

五. (本题 24 分) 试设计一个算法, 改造一个带表头结点的双向链表, 所有结点的原有次序保持在各个结点的右链域 rLink 中, 并利用左链域 lLink 把所有结点按照其值从小到大的顺序连接起来。

第二部分 C 语言程序设计

六. (本题 18 分, 每小题 4.5 分) 阅读以下程序, 回答后面的问题。

```
#include <stdio.h>
int Sub( char *p, char *q )
{
    int i = -1;

    while( p[++i] != '\0' ) {
        if( *(q+i) == '\0' ) return 0;
        if( p[i] == *(q+i) ) continue;
        return 0;
    }
    return 1;
}

void main( )
{
    char buf[256];

    gets( buf );
    while( buf[0] != '\0' ) {
        if( Sub( "http", buf ) )
            printf( "%d\n", buf );
        gets( buf );
    }
}
```

回答以下问题:

- (a) 函数 Sub 的功能是什么? 返回值是什么?
- (b) 变量 buf 的作用是什么?
- (c) 程序的功能是什么?
- (d) 程序中存在错误, 请指出错误所在?

北京工业大学 2016 年硕士研究生入学考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

七. (本题 32 分) 按照以下设计要求, 用 C 语言编码实现。

- (a) 从键盘读入 1 个整数 M 和 1 个数字 K ;
- (b) 从指定文件 File.txt 中读入所有数字 (假设该文件包含多个数字)。
- (c) 找到并输出倒数第 M 个大于 K 的数字; 如果没有找到, 则输出 -1。
- (d) 要求程序中至多保存 $M+1$ 个数字。

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

★所有答案必须做在答题纸上, 做在试题纸上无效

第一部分 数据结构

一、填空题 (本大题共 10 小题, 每小题 2 分, 共 20 分) 不写解答过程, 将正确的答案写在每小题的空格内。错填或不填均无分。

1. 当用二叉链表作为 n 个结点的二叉树的存储结构时, 空指针域的个数是_____。
2. 设双向循环链表中结点的结构为 (data, prior, next), 在 p 所指结点的后面插入一个新结点 $*t$, 需要修改四个指针, 分别为_____。
3. 对用邻接矩阵表示的图进行任一种遍历时, 其时间复杂度为_____, 对用邻接表表示的图进行任一种遍历时, 其时间复杂度为_____。
4. 设栈 S 和队列 Q 的初始状态为空, 元素 e_1 、 e_2 、 e_3 、 e_4 、 e_5 和 e_6 依次通过栈 S , 一个元素出栈后即进入队列 Q , 若 6 个元素出列的顺序是 e_2 、 e_4 、 e_3 、 e_6 、 e_5 、 e_1 , 则栈 S 的容量至少应该是_____。
5. 若对长度 $n = 10000$ 的线性表进行二级索引存储, 每级索引表中的索引项是下一级 20 个表项的索引, 则一级索引表的长度为_____, 二级索引表的长度为_____。
6. 当向一个最小堆插入一个具有最小值的元素时, 该元素需要逐层_____调整, 直到被调整到_____位置为止。
7. 解决散列法中出现的冲突问题的方法有_____和_____。
8. 快速排序的平均时间复杂度为_____, 最坏时间复杂度为_____。
9. 在按层次遍历二叉树的算法中, 需要借助的辅助数据结构是_____。
10. 一维数组的逻辑结构是_____, 存储结构是_____。

二、简答题 (本大题共 4 小题, 每小题 6 分, 共 24 分)

1. 对于如图1所示的二叉树, 请: (1) 画出其顺序存储结构图。(2) 写出该二叉树的前序序列、中序序列和后序序列。

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

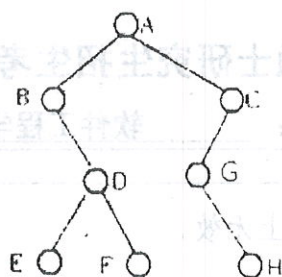


图 1

- 画出对长度为 10 的有序表进行折半查找的判定树, 并求其等概率时查找成功的平均查找长度。
- 顺序队列的“假溢出”是怎样产生的? 如何知道循环队列是空还是满?
- 指出算法 fun 的功能并求出其时间复杂度。

```

int fun (int n) {
    int i = 1, s = 1;
    while (s < n) s += ++i;
    return i;
}

```

三. (本题 16 分) 设有一个关键字的输入序列 {55, 31, 11, 37, 46, 73, 63, 02, 07}, 试:

- 从空树开始构造平衡二叉搜索树, 画出每加入一个新结点时二叉树的形态, 若发生不平衡, 指明需做的平衡旋转的类型及平衡旋转的结果。
- 计算该平衡二叉搜索树在等概率下的搜索成功的平均搜索长度和搜索不成功的平均搜索长度。

四. (本题 20 分) 请对下面的生产计划问题进行分析, 然后建立该问题的数学模型。

生产计划问题: 某工厂生产甲、乙两种产品, 甲产品每生产一件需消耗黄铜 2kg, 3 个工作日, 两个外协件, 每件可获利润 60 元; 乙产品每生产一件需消耗黄铜 4kg, 1 个工作日, 不需要外协件, 每件可获利润 30 元, 该厂每月可供生产用的黄铜 320kg, 总工作日 180 个, 外协件 100 个, 问怎样安排生产才能使工厂的利润最高?

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

五. (本题20分) 假设有向图采用邻接表存储, 设计一个算法, 判定该图中是否存在回路。

- (1) 请概要说明你的算法思想。
- (2) 编写你的算法, 请在算法关键的地方给出必要的注释。
- (3) 分析该算法的时间复杂度。

第二部分 C 语言程序设计

六. 单项选择题 (本大题共 5 小题, 每小题 2 分, 共 10 分)

1. 以下不是死循环的语句为 ()。

- A. for(; ; x+=k);
- B. while(1){x++};
- C. for(k=10; ; k--) sum+=k;
- D. for(; (c=getchar())!='\n';) printf("%c",c);

2. 有以下程序段:

```
main()
{
    int i=1,sum=0,n;
    scanf("%d",&n);
    do
    {
        i+=2;
        sum+=i;
    }while(i!=n);
    printf("%d",sum);
}
```

若使程序的输出值为 8, 则应该从键盘输入的 n 的值是: ()。

- A. 1
- B. 3
- C. 5
- D. 7

3. 在 C 语言中, 逻辑运算符的优先级从高到低的排列顺序为()。

- A. &&、||、!
- B. ||、&&、!
- C. &&、!、||
- D. !、&&、||

4. 在下列选项中, 不正确的赋值语句是()。

- A. ++t;
- B. n1=(n2=(n3=0));
- C. k=i=j;
- D. a=b+c=1;

5. 以下错误的描述是 ()。

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

- A. 不同的函数中可以使用相同的变量名
 B. 形式参数是局部变量
 C. 一个函数内部定义的变量只能在本函数范围内有效
 D. 在一个函数内部的复合语句中定义的变量可以在本函数范围内有效

七. 程序填空 (本大题共 2 小题, 共 14 分)

1. (8 分)

功能: 数组名作为函数参数, 求平均成绩。

/* 定义求平均值函数, 形参为一浮点型数组名 */

```
float aver(float a[ ])
{
    int i;
    float av, s=a[0];
    for(i=1; i<5; i++)
        s+=【①】[i];
    av=s/5;
    return 【②】;
}

void main()
{
    float sco[5], av;
    int i;
    printf("input 5 scores: n");
    for(i=0; i<5; i++)
        scanf("%f", 【③】);
    av=aver(【④】);
    printf("average score is %5.2f\n", av);
    getch();
}
```

①

②

③

④

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

2. (6 分)

/*-----

功能: 统计子字符串 substr 在字符串 str 中出现的次数。例如: 若字符串为"aaas lkaaas", 子字符串为"as", 则应输出 2。若字符串为"asasasa", 子字符串为"asa", 则应输出 3。

-----*/

#include <stdio.h>

fun (char *substr, char *str)

{

int i, j, k, num=0;

for(i=0; str[i]; i++)

for(j=i, k=0; substr[k]==str[j]; k++, j++)

if(substr[①]=='\0')

{

num++;

② ;

}

return num;

}

main() {

char str[80], substr[80];

printf("Input a string:");

gets(str);

printf("Input a substring:");

gets(substr);

printf("%d\n", ③);

}

① _____ ② _____ ③ _____

八. 读程序题 (一共 3 个题目, 共计 10 分)

1. 读下面程序, 给出运行结果 (2 分)。

#include <stdio.h>

int fun()

北京工业大学 2017 年硕士研究生招生考试试题

科目代码:

893

科目名称:

软件工程学科专业基础

```

{
    int x=1;
    static int y=1;
    x--;
    y++;
    return x+y;
}

main()
{
    int a,b;
    a=fun();
    b=fun();
    printf("%d,%d\n",a,b);
}

```

2 读下面程序, 给出运行结果 (2 分)。

```

#include<stdio.h>

main()
{
    int a,b,c,d;
    a=c=0;
    b=1;
    d=20;
    if(a)
        d=d-10;
    else if(!b)
        if(!c)
            d=25;
    else
        d=15;
    printf("d=%d\n",d);
}

```

北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

3. 读下面程序, 回答问题 (6 分)。

```
#include <stdio.h>
#define LEN sizeof(struct student)
struct student {
    long num;
    struct student *next;
};
struct student *creat() {
    struct student *head=NULL,*tail;
    long num;
    int a;
    tail=(struct student *)malloc(LEN);
    printf("please input date\n");
    do
    {
        scanf("%ld",&num);
        if(num!=0)
        {
            if(head==NULL)
                head=tail;
            else
                tail=tail->next;
            tail->num=num;
            tail->next=(struct student *)malloc(LEN);
        }else
            tail->next=NULL;
    }while(num!=0);
    return(head);
}
main() {
    struct student *p;
    p=creat();
```


北京工业大学 2017 年硕士研究生招生考试试题

科目代码: 893 科目名称: 软件工程学科专业基础

printf("you input is n");

while(p)

{

printf("%d\n", p->num);

p=p->next;

}

}

回答以下问题:

(a) 函数 creat 的功能是什么? 返回值是什么?

(b) 程序的功能是什么?

(c) num 为 0 时程序有什么行为?

九. 编程题 (一共 2 个题目, 共计 16 分)

1. 编程判断输入的字符串是否为“回文”。所谓“回文”是指顺读和倒读都一样, 如 abcba, (8 分)

2. 编程计算 $1! + 2! + 3! + 4! + \dots + 10!$ 的值, (8 分)