

# 华中科技大学

## 二〇〇五招收硕士研究生入学考试试题

考试科目: 数据结构及程序设计技术

适用专业: 生物信息技术

(除画图题外, 所有答案都必须写在答题纸上, 写在试题上及草稿纸上无效, 考完后试题随答题纸交回)

### 数据结构部分 (共 80 分)

#### 一、选择题 (每空 1 分, 共 10 分)

- 对任意的 7 个关键字进行排序, 至少要进行\_\_\_\_\_次关键字之间的两两比较。  
A. 16    B. 15    C. 14    D. 13
- 下列\_\_\_\_\_是线性表。  
A. {A, B, C, D, E, F}    B. (GO0005216, GO0004317, GO0002463, GO0050007)  
C. (0, 1, 2, 3, 4, ...)    D. ((1), (2, 3), (4, 5, 6))
- 设栈的输入序列是 1, 2, 3, 4, 则\_\_\_\_\_不可能是其出栈序列。  
A. 1432    B. 2134    C. 1423    D. 4321
- 假定一个顺序循环队列的队首和队尾指针分别用 *front* 和 *rear* 表示, 则\_\_\_\_\_是判队空的条件。  
A.  $front + 1 == rear$     B.  $front == rear + 1$   
C.  $front == 0$     D.  $front == rear$
- 若串  $S = "BIOINFORMATICS"$ , 其子串的数目是\_\_\_\_\_。  
A. 14    B. 105    C. 98    D. 95
- 若用单链表来表示队列, 则应该选用\_\_\_\_\_。  
A. 带尾指针的非循环链表    B. 带尾指针的循环链表  
C. 带头指针的非循环链表    D. 带头指针的循环链表
- 若需在  $O(n \log_2 n)$  的时间内完成对数组的排序, 且要求排序是稳定的, 则可选的排序方法是\_\_\_\_\_。  
A. 快速排序    B. 堆排序    C. 归并排序    D. 直接插入排序

试卷编号: 440

共 6 页  
第 1 页

准考证号码:

题  
写  
要  
不  
内  
线  
密  
封

专业:  
报考科:

姓名:

8. 在一个具有  $n$  个结点的有序单链表中插入一个新结点并仍然保持有序的时间复杂度是\_\_\_\_\_。
- A.  $O(1)$       B.  $O(n)$       C.  $O(n^2)$       D.  $O(n\log_2 n)$
9. 一个  $n \times n$  的对称矩阵, 如果以行或列为主序放入内存, 则容量为\_\_\_\_\_。
- A.  $n \times n$       B.  $n \times n/2$       C.  $n \times (n+1)/2$       D.  $(n+1) \times (n+1)/2$
10. 如果要求一个线性表既能较快地查找, 又能适应动态变化的要求, 可以采用\_\_\_\_\_查找方法。
- A. 分块      B. 顺序      C. 折半      D. 散列

## 二、简单计算 (每小题 6 分, 共 24 分)

1. 试用关键字(19,25,37,4,6,56,34,26,67,9)构造 Hash 表, Hash 函数为  $h(key) = key \bmod 11$ , 其中  $key$  为关键字,  $mod$  为取模运算。用开地址法处理冲突, 线性探测再散列法查找空位, Hash 表的长度为 12。要求: (1)画出 Hash 表的存储结构图; (2)假定每个元素的查找概率相同, 计算查找成功时的平均查找长度。
2. 试说明归并排序的基本过程, 并给出对数组 {21,14,52,46,13,78,91,55,43} 进行两路归并排序的过程。
3. 试用权 {6,8,7,10,2,4,3} 构造 Huffman 树, 要求: (1)列出构造过程; (2)计算树的路径长度; (3)计算 Huffman 树的带权路径长度。
4. 考虑图 1 中的网  $G$ , 从顶点  $A$  出发, 用 Prim 算法求一棵最小生成树, 列出主要的求解过程。

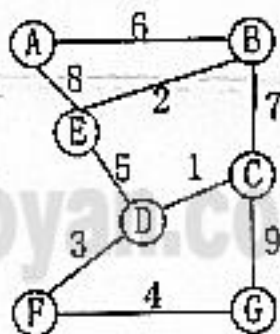


图1

## 三、分析和证明 (每小题 6 分, 共 18 分)

1. 试分析用线性表的顺序存储结构来描述蛋白质序列的序列、结构域、引用文献、对应的基因序列、相关疾病特征等相关信息是否合适? 为什么?
2. 设  $G$  为有  $n$  个顶点的无向连通图, 证明  $G$  至少有  $n-1$  条边。
3. 证明只要适当地排列顶点的次序就能使无环有向图的关系矩阵中对角线以下的元素全为 0。

## 四、算法设计 (每小题 14 分, 共 28 分)

1. 假定采用顺序结构存储串, 设计一个算法, 求串  $s$  中出现的第一个最长重复子串及其位置, 给出算法的时间复杂度。

2. 设二叉树的结点结构是:

Left Child	Data	Right Child
------------	------	-------------

其中 *Left Child* 和 *Right Child* 分别为指向左、右子树根的指针, *Data* 是字符串数据。试写出算法, 求任意二叉树中第一条最长的路径长度, 并输出此路径上各结点的值。

## C 语言程序设计部分 (共 70 分)

一、简答题 (每小题 2 分, 共 16 分)

1. 给出“*f* 是返回值为无符号指针的无参函数”的定义。
2. 说明在一个源文件中定义的全局变量的作用域范围。
3. 打开文件, 对其操作完毕之后, 应该怎样操作才能避免数据丢失。
4. 写出“如果变量 *d* 是小写字母, 则将 *d* 转换成大写字母, 否则 *d* 的值不变”的表达式。
5. 写出“使得 *x* 的高 3 位不变, 低 5 位取反 (*x* 为字符型变量)”的表达式。
6. 写出“*x, y* 中存在有一个小于 *z* (*x, y, z* 均为 *int* 型变量)”的表达式。
7. 写出“将 *int* 型变量 *x* 缩小 2 的 *n* 次方倍”的表达式。
8. 设有说明语句 `char s[40]`; 如果想从终端上把字符串“*Associates with chromatin.*”送到数组 *s* 中, 应当调用什么函数, 写出其调用形式。

二、写出各程序的输出结果 (每小题 4 分, 共 40 分)

```
1. #include <stdio.h>
   main()
   {   int a=-32768;
       unsigned int b;
       b=a;
       printf("a=%d,b=%u\n",a,b);
   }
```

```
2. #include <stdio.h>
   void sub(int *x, int n, int k)
   {   if (k<=n) sub(x,n,2*k);
       *x += k;
   }
   void main()
   {   int x=0;
       sub(&x,10,1);
       printf("x=%d\n",x);
   }
```

```

3. #include <stdio.h>
void swap(int *p1, int *p2)
{   int p;
    p=*p1; *p1=*p2; *p2=p;
}
void main()
{   int a,b;
    int *point1, *point2;
    scanf("%d,%d",&a,&b);
    point1=&a; point2=&b;
    if (a>b) swap(point1,point2);
    printf("%d,%d",a,b);
}

```

若输入 8, 3, 则输出结果为\_\_\_\_\_

```

4. #include <stdio.h>
int f1(int x, int y, int *sum)
{   *sum = x+y;
    ++x; ++y;
}
void main()
{   int f1(), (*f)0;
    int a=40, b=50, c=80;
    f=f1;
    (*f)(a,b,&c);
    printf("%d,%d,%d\n",a,b,c);
}

```

```

5. void main()
{   struct { int a; int * b; } s[4], *p;
    int n=10, i;
    for (i=0; i<4; i++) {
        s[i].a=n++;
        s[i].b=&s[i].a;
    }
    p=&s[1];
    printf("%d,", (++p)->a);
    printf("%d\n", (p++)->a);
}

```

```
6. #include <stdio.h>
void main()
{   int a[]={2,4,6}, *ptr=&a[0], x=8, y, z;
    for (y=0; y<3; y++) z=*(ptr+y)<x?*(ptr+y):x;
    printf("%d\n",z);
}
```

---

```
7. #include <stdio.h>
void main()
{   char str[]="213\0123\0321", * p = str;
    printf("%s",p+5);
}
```

---

```
8. #include <stdio.h>
void main()
{   int a[2][2]={1,2,3,4}, i, j, s;
    for (s=0,i=0; i<2; i++)
        for (j=0; j<2; j++) s = s + a[0][i*2+j];
    printf("s=%d\n",s);
}
```

---

```
9. #include <stdio.h>
void main()
{   char *a[]={"gene", "protein", "structure"};
    char *p;
    for ( p=a; p<a+3; p++) printf("%c,", *(p+3));
}
```

---

```
10. #include <stdio.h>
void main()
{   int x,y;
    for (y=1, x=1; y<=50; y++) {
        if (x == 10) break;
        if (x%2) { x+=5; continue; }
        x-=3;
    }
    printf("x=%d,y=%d",x,y);
}
```

---

三、完善程序（每空 2 分，共 14 分）

1. 下面程序的功能是在结构体数组中查找 *score* 值最高和最低的 *protein* 名称和分值。

```
#include <stdio.h>
main()
{   int max,min,i,j;
    static struct
    {   char name[6];
        int score;
    }proseq[5]={“O00168”,13,“O00134”,11,“P00238”,9,“P00213”,1,“Q00016”,18};
    max=min=0;
    for (i=1; i<5; i++)
        if (proseq[i].score>proseq[max].score) (1) ;
        else if (proseq[i].score<proseq[min].score) (2) ;
    printf(“high score protein sequences: %s,%d\n”, (3));
    printf(“low score protein sequences: %s,%d\n”, (4));
}
```

2. 下面程序的功能是用折半法进行查找。

```
int binsearch(int x, int v[], int n)
{   int low, high, mid;
    low = 0;
    high=n-1;
    mid=(low+high)/2;
    while ( (1) ) {
        if ( x<v[mid] ) high=mid-1;
        else low=mid+1;
        (2);
    }
    if ( (3) ) return (mid);
    else return (-1);
}
```