

中山大学

二〇一〇年攻读硕士学位研究生入学考试试题

科目代码: 872

科目名称: 数据结构 (A)

考试时间: 1 月 10 日 下 午

考生须知

全部答案一律写在答题纸上,
答在试题纸上的不得分! 请用蓝、
黑色墨水笔或圆珠笔作答。答题要
写清题号, 不必抄原题。

注意: 请把答案按顺序写在答题纸上, 并标明题号。

一. 下面列出某些对象所具备的特性:

- (1) 数据的存储顺序可与逻辑顺序不一致. (2) 插入和删除操作时涉及元素移动.
(3) 常设置首指针和尾指针. (4) 常用于实现将数据的顺序颠倒. (5) 数据项中附加了指针字段. (6) 排序后可用二分法查找. (7) **push, pop** 是其典型操作.
(8) 先进先出. (9) 所占用的空间往往多于实际有效数据所需要的空间. (10) 操作时涉及内存的申请与释放. (11) 先进后出. (12) 可模拟排队现象.

问题 1: 对于下面每一种对象, 选择三个最明显的特征, 写出其序号. (10 分)

- ▽ 基于数组的顺序表: _____.
▽ 单链表: _____.
▽ 堆栈: _____.
▽ 队列: _____.

问题 2: 请为上述对象各增加一个特征或优点. (10 分)

二. 下面给出一些关键词: MaxSize, Length, data, next, top, front, rear, lchild, rchild, prior.

问题 1: 选择上述关键词将下面的定义补充完整, 使得对象的定义具有很好的直观性. (15 分)

(1) 定义线性表

```
Typedef struct  
{ ElemType data[_____];  
  int _____;  
} SqList;
```

(3) 定义顺序式队列:

```
Typedef struct  
{ ElemType data[_____];  
  int _____, _____;  
} SqQueue;
```

(2) 定义顺序式堆栈

```
Typedef struct  
{ ElemType data[_____];  
  int _____;  
} SqStack;
```

(4) 定义单链表:

```
Typedef struct LNode  
{ ElemType _____;  
  struct LNode * _____;  
} LinkList;
```

考试完毕, 试题和草稿纸随答题纸一起交回。

第 1 页 共 4 页

(5) 定义双链表:

```
typedef struct DNode
{ ElemType _____ ;
  struct DNode * _____ , * _____ ;
} DLinkedList;
```

(6) 定义二叉树

```
typedef struct Node
{ ElemType _____ ;
  struct Node * _____ , * _____ ;
} BTNode ;
```

问题 2: 请给出下列对象的定义.

(15 分)

- (1) 稀疏矩阵的三元组表示.
- (2) 多项式的存储.
- (3) 三叉树. (4) 存储图的邻接矩阵.

三: 有关队列的问题, 设要处理的信息类型为 ElemType:

(25 分)

问题 1: 下列操作中哪些适合于用队列来实现:

子程序调用; 层次遍历一棵二叉树; 后缀表达式 (在字符串 EXP 中) 的计算.

问题 2: 设用长度为 N 的数组 Q 作为循环队列的存储空间, 头尾指针分别为 front, rear: 试给出相关的定义并进行初始化; 画出空队列, 满队列及中间态三种情形下的空间结构示意图.

问题 3: 写出 Queue_In 和 Queue_Out 的实现代码. (入队和出队)

四. 下面列出若干表达:

(25 分)

- (1) 有 prim 算法. (2) 具有邻接矩阵表示法. (3) 具有文氏图表示法. (4) 可用于回溯法. (5) 有种特殊情形称为“平衡的”. (6) 具有先根遍历法. (7) 用于存储数据时可加快查找速度. (8) 往往要考虑连通性. (9) 具有“生成树”. (10) 具有头结点和尾结点. (11) 可以表示“解空间”. (12) 可用于“并查集”. (13) 有一种特殊情形称为“完全的”. (14) 可用于生成哈夫曼编码. (15) 具有根结点. (16) 可用于堆排序. (17) 可用于拓朴排序. (18) 可用于直接插入排序.

问题: 哪些表达适合于“树”, 哪些表达适合于“图”, 请列出表达的序号(列序号时注意由小到大).

1. 适合于“树”的表达: _____.
2. 适合于“图”的表达: _____.

五. 有关时间复杂度的问题:

(20 分)

问题 1: 下面给出函数 fun() 的实现代码, 设给定表长 n, 而 k 是待检索的数, 当 k 取什么值时时间复杂度达到最小; 当 k 取什么值时时间复杂度达到最大; 推算出相应的时间复杂度表达式并推算出平均的时间复杂度表达式(设 k 取表中任何值的概率相等).

```
void fun(int a[ ], int n, int k)
{ int i;
  i = 0;
```

```

        while ( i<n && a[i] != k)
            i ++;
        return(i);
    }

```

问题 2: 下面给出若干时间复杂度的度量函数, 请按渐近阶由低到高排序:

(1) $4n^2$; (2) $\log n$; (3) 3^n ; (4) $20n$; (5) 2 ; (6) $n^{3/2}$; (7) $n!$

问题 3: 设规模为 n 时的复杂性 $T(n)$ 存在下列递推关系:

$$T(n) \begin{cases} = O(1) & (n=1 \text{ 时}) \\ = k T(n/m) + f(n) & (n>1 \text{ 时}) \end{cases}$$

试推导 $T(n)$ 的非递推形式的解的表达式。

六. 代码填空题.

(1) 直接插入排序

(10 分)

```

void StrainSert( ElemType r[], int n )
{
    int i, j;
    ElemType x;
    for(i=_____; i<n; i++)
    {
        x=_____ ; j=_____ ;
        while ( x<r[j] && j>=0 )
        {
            _____ ;
            _____ ;
        }
        _____ = x;
    }
}

```

(2) 起泡排序

(10 分)

```

void BubbleSort( ElemType a[], int n)
{
    ElemType x;
    int i, j, k, flag;
    for(i=0; i<n-1; i++)
    {
        flag=_____ ; // there is any exchange?
        for (j=n-1; j>i; j--)
            if ( _____ )
            {
                // exchange!
            }
    }
}

```

```

        flag = _____ ;
    }
    if ( _____ ) return;
}
}

```

(3) 二分法检索的递归形式

(10 分)

int BinSch(struct Node A[], int low, int high, KeyType K)

```

{ if ( _____ ) // whether continue?

{ int mid=_____ ;
  if ( k == A[mid].key ) return _____ ;
                                // success!
  else if ( _____ )
                                // then go left
    return _____ ;

  else
                                // then go right
    _____ ;
}
else return -1; // fail!
}

```