

南京航空航天大学

第1页 (共7页)

二〇二一~二〇二二学年 第2学期 《数据结构 C#》考试试题

考试日期: 2022 年 5 月 22 日 试卷类型: A 试卷代号:

班号			学号				姓名				
题号	一	二	三	四	五	六	七	八	九	十	总分
得分											

本题分数	20
得分	

一、概念题 (20%)

1. 设有关键字序列{6,5,10,12,3,8,9,7},构造二叉查找树 (10%)

2. 设有关键字序列{2,3,6,8,9,12,15},构造 HASH 表,用溢出表法解决冲突问题,要求给出构造过程 (10%)

$$i=f(k)=\text{int}(k/3)+1$$

本题分数	80
得 分	

二、算法题，仔细阅读附件结构定义，且不得修改定义（80%）

1.用 C 语言写出压栈的算法（10%）

```
int stack_push(stack *s,data *d){
```

```
}
```

2.用 C 语言写出单项链表删除链表头的算法(10%)

```
int slist_delete_head(slist *l){
```

```
}
```

3. 用 C 语言写出双向链表删除节点的算法（10%）

```
int dlist_delete_node(dlist *l,int id){
```

```
}
```

4. 用 C 语言写出设用双向链表构造某课程成绩模型，数据部分为学号和成绩两部分：{int id,float score}，请给出计算并返回平均成绩的算法(10%)

```
float dlist_calc_averge(dlist *l){
```

```
}
```

5. 用 C 语言写出二叉树先序递归遍历，显示节点 id 算法（10%）

```
void btree_iterate(bnode *n){
```

```
}
```

6. 用 C 语言写出树非递归求下一节点的算法（10%）

```
node *tree_get_next_node(node *n){
```

```
}
```

7. 用 C 语言写出有向图求某作业所有紧前作业个数的算法（10%）

```
int pert_get_predecessors(pert *t,int id){
```

```
}
```

8. 用 C 语言写出树中判定两个节点是否存在父子关系的算法（查找节点可以直接使用 tree_find_node 函数）（10%）

```
int tree_has_as_parent(tree *t,int id,int pid){
```

```
}
```

相关数据结构定义:

1、数据

```
typedef struct _data{
    int id ;
    char name[10] ;
}data ;
```

2、单向链表

```
typedef struct _snode{
    data *d ;
    struct _snode *next ;
}snode ;
```

```
typedef struct _slist{
    snode *head,*tail;
}slist ;
```

2、双向链表

```
typedef struct _dnode{
    data *d ;
    struct _dnode *next,*prev;
}dnode ;
```

```
typedef struct _dlist{
    dnode *head,*tail ;
}dlist ;
```

3、二叉树

```
typedef struct _bnode{
    data *d ;
    struct _bnode *left,*right ;
}bnode ;
```

```
typedef struct _btree{
    bnode *root ;
}btree ;
```

4、树

```
typedef struct _tnode{
    data *d ;
    struct _tnode *parent,*first_child,*last_child,*prev_sibling,*next_sibling ;
}bnode ;
```

```
typedef struct _tree{
    tnode *root ;
}tree;
```

5.图

```
typedef struct _data{
```

```
int id;
char name[20];
int duration;
int tes;
int tef;
int tls;
int tlf;
}data;
typedef struct _relation{
    data *d;
    struct _relation *next;
}relation;

typedef struct _pnode{
    data *d;
    relation *predecessor,*successor;
    struct _pnode *next,*prev;
}pnode;

typedef struct _pert{
    pnode *head,*tail;
}pert;
```

6. 栈

```
typedef struct _stack{
    int count;
    int max;
    data element[100];
}stack;
```

7. 队列

```
typedef struct _queue{
    int count;
    int max;
    data element[100];
}queue;
```