

南京航空航天大学

第1页 (共9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 日 试卷类型: B 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	40
得分	

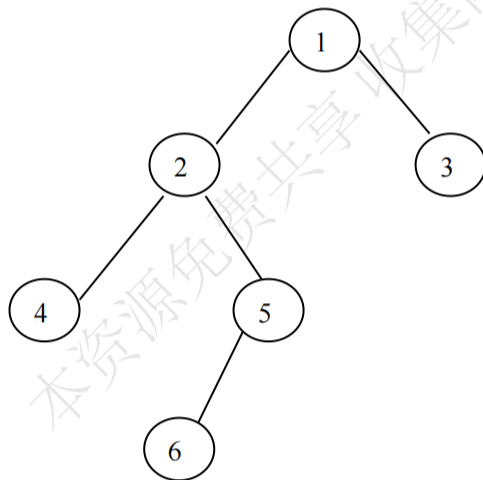
一、基本理论与方法 (每题 10 分, 共 40 分)

1、有序数字序列{1, 2, 5, 7, 9, 10, 15, 20, 28}, 写出用折半查找查找 5 的过程。

2、设有数字序列{12, 8, 1, 3, 5, 10, 7, 22}, 写出快速排序一次快排的过程及结果 (升序)

3、设有关键字系列{1, 3, 4, 8, 9, 13}, 请构造 HASH 表, HASH 函数: $i = (\text{int})(\text{key}/3) + 1$, 用溢出表解决冲突问题。

4、二叉树如图所示, 给出前序、中序和后序遍历的结果



前序:

中序:

后序:

本题分数	60
得 分	

二、 程序题，仔细阅读附件中的结构定义，按要求
写出函数体，(每题 10 分，共 60 分)

1、用 C 语言写出计算单向链表节点个数的算法

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

```
}
```

2、用 C 语言写出双向链表插入节点的算法(插在 id 的后面)

```
int dlist_insert_after(dlist *l, int id, data *d){
```

```
}
```

3、用 C 语言写出二叉树中序遍历的算法，显示节点 id。

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

```
}
```

4、设用链表来实现队列结构，请用 c 语言给出出队操作的算法

```
qnode *queue_leave(queue *q){
```

```
}
```

5、给出计算 BOM 树中零件清单的算法，显示零件的 ID。

```
void *tree_list_parts(tnode *n){
```

```
}
```

6. 写出显示网络图中所有关键作业 ID, 同时返回关键作业的个数计算方法

```
int pert_get_critical_tasks(pert *p){
```

```
}
```

附件：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、栈

```
typedef struct _snode{
```

```
    data *d;

    struct _snode *next,*prev;
}qnode;
typedef struct _stack{
    snode *head,*tail ;
}stack ;
```

4、队列

```
typedef struct _qnode{
    data *d;

    struct _qnode *next,*prev;
}qnode;
typedef struct _queue{
    qnode *head,*tail ;
}queue ;
```

5、二叉树

```
typedef struct _bnode{
    data *d ;

    struct _bnode *parent,*left,*right ;
}bnode ;
typedef struct _btree{
    bnode *root ;
}btree ;
```

6、树

```
typedef struct _tnode{
    data *d;
```

```

    struct _tnode *parent;

    struct _tnode *first_child,*last_child;

    struct _tnode *prev_sibling,*next_sibling;
}tnode;
typedef struct _tree{
    tnode *root;
}tree;

```

7、图

```

typedef struct _data{
    int id;
    char name[20];
    int duration;        //周期
    int tes;              //最早开始时间
    int tef;              //最早完工时间
    int tls;              //最迟开始时间
    int tlf;              //最迟完工时间
    int critical;         //关键作业
    int done;             //结束标记
}data;
typedef struct _relation{
    data *d;
    struct _relation *prev,*next;
}relation;
typedef struct _task{
    data *d;
    relation *predecessor,*successor;
    struct _task *prev,*next;
}

```



```
}task;  
typedef struct _pert{  
    int id;  
    char name[20];  
    int tes;    //项目最早开始时间  
    int tlf;    //项目合同完工时间  
    task *head,*tail;  
}pert;
```

本资源免费共享 收集网站 nuaa.store