

1993 年上海交通大学数据结构及程序设计技术试题

考研加油站收集整理 <http://www.kaoyan.com>

1993 年上海交通大学数据结构及程序设计技术试题



上海交通大学 1993-

年研究生入学考试试题

试题名称: 数据结构及程序设计技术

题一(14分) 用循环单链表实现队列操作。已给定下列定义和说明:

```
type datatype = integer; {结点中的数据类型}
    pointer = ↑ node;
    node = record data: datatype; next: pointer end;
var tail: pointer; {只用尾指针,不设首指针}
procedure clear; begin tail := nil end;
{clear用于将队列初始化;当队列空时,要求clear.}
```

试用pascal语言写出进队操作 enqueue 和出队操作 dequeue. 它们的过程首部分别给定为:

```
procedure enqueue (x: datatype);
procedure dequeue (var x: datatype);
```

题二(11分) 已知某有序树的前序遍历序列为 ABCDEFGH, 后序遍历序列为 BDFEGCHA. 试画出这棵有序树图形, 并说明画图方法.

题三(12分) 设分类(排序)二叉树结点类型定义如下:

```
type Keytype = integer; {关键字类型}
    pointer = ↑ node;
    node = record key: keytype; left, right: pointer end;
```

阅读下列用pascal语言写出的删除分类二叉树结点 tkey 的过程说明. 试在答卷纸上写出程序中各 i 的编号 i 和它对应的表达式, 以完成该段程序.

```
procedure deletekey (tkey: keytype; var fail: boolean);
var p, remove: pointer;

procedure subdel (var q: pointer); {找替代结点}
```

```

begin { subdel }
  if  $q \uparrow .right \neq nil$  then subdel ( 1 )
  else begin remove  $\uparrow .key :=$  2 ;
        remove :=  $q$  ;
         $q :=$  3
      end
end { subdel } ;

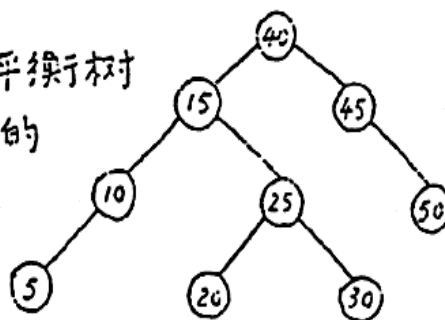
procedure del ( tkey : keytype ; var p : pointer ) ;
  { 寻找被删结点 remove }
begin { del }
  if  $p = nil$  then fail := true    { 找不到, 失败 }
  else if  $tkey < p \uparrow .key$  then del ( tkey, 4 )
  else if  $tkey > p \uparrow .key$  then del ( tkey, 5 )
  else begin { 找到 }
        remove :=  $p$  ; fail := false ;
        { 以下为替代 (包含删除) }
        if  $p \uparrow .right = nil$  then  $p :=$  6
        else if  $p \uparrow .left = nil$  then  $p :=$  7
        else subdel (  $p \uparrow .left$  )
      end
  end { del } ;

begin { deletekey }
  del ( tkey, root ) { root 为被删分类二叉树的根指针 }
  if 8 then dispose ( remove )
end

```

其中, 方框 1, 4, 5 为变量, 其它方框都是表达式。

题四(10分) 试画出右边平衡树
插入结点 35 及调整后的
平衡树图形。
(只要求画图)



题五(14分) 设无向连通图G如下图左所示,它的邻接表见下图右所示。设邻接表已按下面定义的数据类型存入计算机中。

```
const n=6;    {n为图G的顶点个数}
type keytype = 1..n;    {顶点关键字类型}
    pointer = ↑ enode;
    enode = record key: keytype; link: pointer end;
var V: array[keytype] of pointer;
```

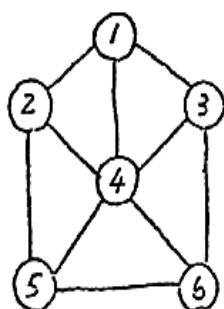
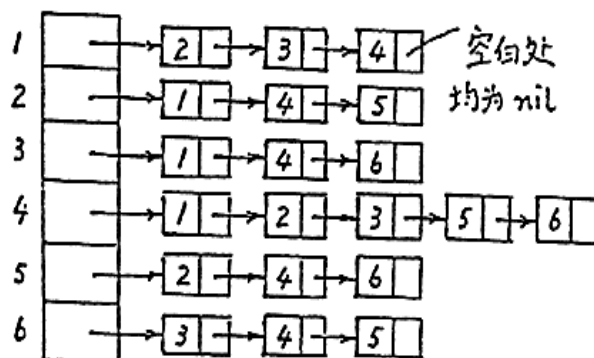


图 G



数组 V

结点类型为 enode

下面是用 pascal 语言写出的对图进行某种方式搜索的过程说明。

```
procedure graph-search;
const m=20;    {m为图的边的个数的2倍}
var a: array[1..m] of keytype;    {a作为栈或队使用}
    i, j: 0..m;    {i, j 均用来存放数组 a 的下标}
    visited: array[keytype] of boolean;
    {visited 初值为 false, 当结点 k 被访问后, visited[k] 为 true}
    k: keytype;    p: pointer;
begin
  for k:=1 to n do visited[k]:=false;
  i:=0; j:=0;
  i:=i+1; a[i]:=1;    {结点 1 进入数组 a}
  while i <> j do
  begin    待填    {将数组 a 中的一个元素退出, 赋给变量 k}
```

```

if not visited[k] then
begin write(k); {访问结点k}
  visited[k]:=true;
  p:=v[k];
  while p<>nil do {将k的邻接结点全部进入数组a}
begin i:=i+1; a[i]:=p↑.key; p:=p↑.link
end
end
end {while}
end

```

试1. 如果希望实现深度优先搜索, 请写出:

- (1) 待填 处应是哪些语句;
- (2) 图G的结点被搜索的顺序;
- (3) 画出相应的生成树。

试2. 如果希望实现宽度优先搜索, 也请写出上面的三个问题的答案。

题六 (12分) 用PASCAL语言写出计算赋权有向图每一个有序顶点对 $\langle i, j \rangle$ 之间最短路径的Floyd算法 (不要写出完整的程序, 只需写出若干语句)。再问: (1) Floyd算法何时无效, (2) 复杂度是多少?

题七 (17分) 下面PASCAL语言的程序中。过程reada 读入数据到数组a中; 过程print 将a中数据输出; 过程heapsort为将数组a中数据按由小到大顺序进行排序 (堆分类算法); 过程reverse将a中数据前后真颠倒; 函数search(k)为在数组中用对分法搜索元素k的位置。

请阅读该段程序。

试1. 写出程序中各 i 的编号i及其对应的内容, 以完成该程序。

试2. 设输入数据为4, 3, 1, 6, 8。写出程序的输出结果。

(第七题程序)

```

program T.7 (input, output);
const n = 5;
var a: array [1..n] of integer;

procedure reada;
var i: 1..n;
begin for i := 1 to n do read(a[i]) end;

procedure print;
var i: 1..n;
begin for i := 1 to n do write(a[i]); writeln end;

procedure heapsort;
var l, r: 1..n; x: integer;
procedure sift; {在数组a从l到r这一段中, 将l调整到}
label 13; {适当位置, 使它仍然是最大化的堆}
var i, j: integer;
begin i := l; j := 2 * i; x := a[i]; {i为父亲, j为左儿子}
  while j <= r do
  begin
    if j < r then if  then j := j + 1; {使j指向大儿子}
    if x > a[j] then goto 13; {已找到适当位置, goto 13}
    ; {调整}
    i := j; j := 2 * i
  end;
  13: a[i] := x
end {sift};
begin {heapsort}
  l := (n div 2) + 1; r := n;
  while l > 1 do ; {建立初始堆}
  print; {输出数组a}
  while r > 1 do
  begin ; {交换}
    r := r - 1; sift; {调整}
    print {输出数组a}
  end
end {heapsort};

```



```

procedrue reverse;
var i: 1..n; x: integer;
begin for i:=1 to n div 2 do
    begin x:=a[i]; a[i]:=a[n+1-i]; a[n+1-i]:=x end
end;

function search(k: integer): integer;
var l, h, m: 1..n; found: boolean;
begin l:=1; h:=n; found:=false;
    while (l <= h) and not found do
        begin m:=(l+h) div 2;
            if k=a[m] then found:=true
            else if k > a[m] then h:= 5
            else l:= 6
        end;
    if found then search:=m else search:=0
end {search}
begin {main}
    reada; heapsort;
    writeln(search(6));
    reverse;
    writeln(search(6))
end.

```

单独考试的考生不做第八题。

题八(10分) 已知完全二叉树(每个结点或者没有儿子,或者有两个儿子)的前序遍历序列和后序遍历序列,是否能唯一确定这棵完全二叉树,即不存在两棵不同的完全二叉树,它们具有相同的前序遍历序列和相同的后序遍历序列。如果结论是肯定的,试证明之;如果结论是否定的,试画出两棵完全二叉树,使它们的前序和后序分别相等。