

## 1999 年上海大学程序设计与数据结构试题

考研加油站收集整理 <http://www.kaoyan.com>

1999 年上海大学程序设计与数据结构试题



上海大学1999 年攻读 硕 士学位研究生

## 入学考试试题

招生专业 计算机系统、计算机组成原理、计算机组成原理 考试科目 程序设计与数据结构

一、请完善下列程序或算法流程图，其中第 1、2、3 小题在

(a), (b) 中任选一题：(共 36 分)

1. 下列的程序段 search(a,n,k)在数组 a 的前  $n(n \geq 1)$  个元素中找出第  $k(1 \leq k \leq n)$  小的值。这里假设数组 a 中各元素的值都不相同。(8 分)

程序 (a)

```
#define MAXN 100
```

```
int a[MAXN], n, k;
```

```
int search_c(int a[], int n, int k)
```

```
{
```

```
    int low, high, i, j, m, t;
```

```
    k--; low=0; high=n-1;
```

```
    do {
```

```
        i=low; j=high; t=a[low];
```

```
        do {
```

```
            while (i<j && t<a[j]) j--;
```

```
            if (i<j) a[i++] = a[j];
```

```
            while (i<j && t>=a[i]) i++;
```

```
            if (i<j) a[j--] = a[i];
```

```
        } while (i<j);
```

```
        a[i]=t;
```

```
        if (_____(1)_____)
```

```
            if (i<k)
```

```
                low= _____(2)_____;
```

```
            else
```

```
                high= _____(3)_____;
```

```
        } while (_____(4)_____);
```

```
    return (a[k]);
```

程序 (b)

```
const maxn = 100;
```

```
type atype = array[1..maxn] of integer;
```

```
var
```

```
    a : atype;
```

```
    n,k,i : integer;
```

```
function search_pascal(var a:atype; n,k: integer):integer;
```

```
var low , high, i, j, t : integer;
```

```
begin
```

```
    low := 1;    high := n;
```

```
    repeat
```

```
        t := a[low];    i := low;    j := high;
```

```
        repeat
```

```
            while (i<j) and (t<a[j]) do
```

```
                j:=j-1;
```

```
            if (i<j) then
```

```
                begin
```

```
                    a[i]:=a[j];
```

```
                    i:=i+1;
```

```
                end;
```

```
            while (i<j) and (a[i]<=t) do
```

```
                i:=i+1;
```

```
            if (i<j) then
```

```
                begin
```

```
                    a[j]:=a[i];
```

```
                    j:=j-1
```

2. 假设 root 是一棵给定的非空查找树, 对于下面给出的子程序, 当执行注释中给出的调用语句时, 就可以实现如下的操作: 在非空查找树 root 中查找值为 k 的结点, 若值为 k 的结点在树中, 且是一个叶子结点, 则删除此叶子结点, 同时置 success 为“真”; 若值为 k 的结点不在树中, 或者虽然在树中, 但不是叶子结点, 则不进行删除, 仅置 success 为“假”。应注意到非空查找树只包含一个结点的情况, 此时树中的唯一结点, 既是根结点, 也是叶子结点。(8 分)

程序 (a)

```
#include <stdio.h>
typedef struct node {
    int key;
    struct node *left, *right;
} NODE;
NODE *root;
int k, success;

void del_leaf(NODE **t, int k, int *sn)
{
    NODE *p, *pf;
    p = *t; *sn = 0;
    while ((1) _____ && !*sn)
        if (k == p->key) *sn = 1;
        else {
            _____ (2) _____;
            if (k < p->key) p = p->left;
            else p = p->right;
        }
    if (*sn && p->left == NULL && p->right == NULL) {
        if (_____ (3) _____)
            if (pf->left == p) pf->left = NULL;
            else pf->right = NULL;
        else _____ (4) _____;
        free(p);
    }
}
```



程序 (b)

```
const maxn = 50;
```

```
type    datatype = integer;
        nodeptr = ^nodetype;
        nodetype = record
                                key: datatype;
                                left, right : nodeptr
                            end;
```

```
var root: nodeptr;    k: integer;    success: boolean;
```

```
procedure del_leaf(var t: nodeptr; k: integer; var sn: boolean);
```

```
var p, pf: nodeptr;
```

```
begin
```

```
  p := t;    sn := false;
```

```
  while (_____(1)_____) and not sn do
```

```
    if (k = p^.key) then sn := true;
```

```
    else begin
```

```
      _____(2)_____;
```

```
kaoyan.com if (k < p^.key) then
```

```
  p := p^.left;
```

```
  else
```

```
    p := p^.right;
```

```
  end;
```

```
  if sn and (p^.left = nil) and (p^.right = nil) then
```

```
    begin
```

```
      if (_____(3)_____) then
```

```
        if (pf^.left = p) then pf^.left := nil
```

```
        else pf^.right := nil
```

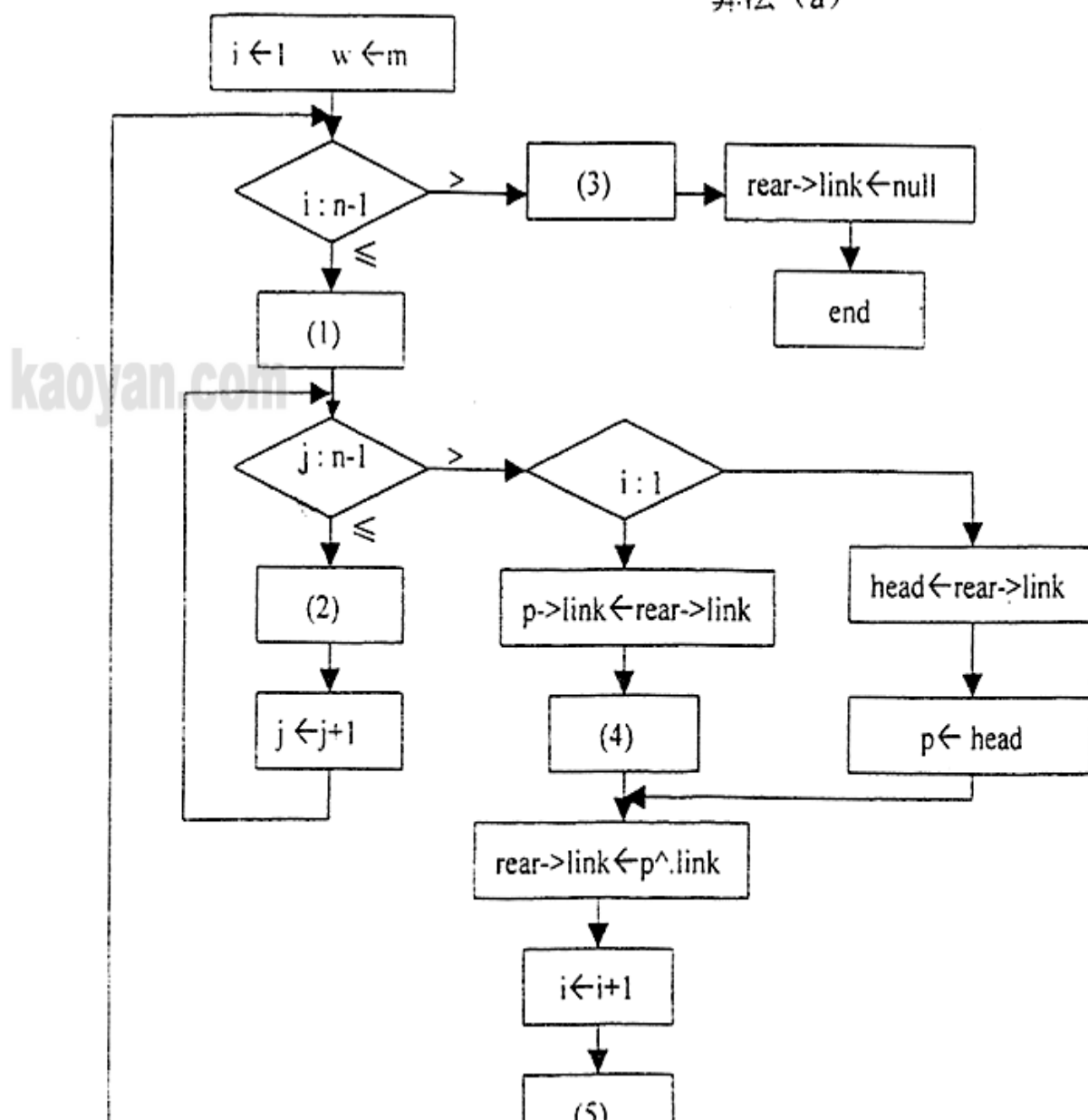
```
      else _____(4)_____;
```

```
      dispose(p);
```

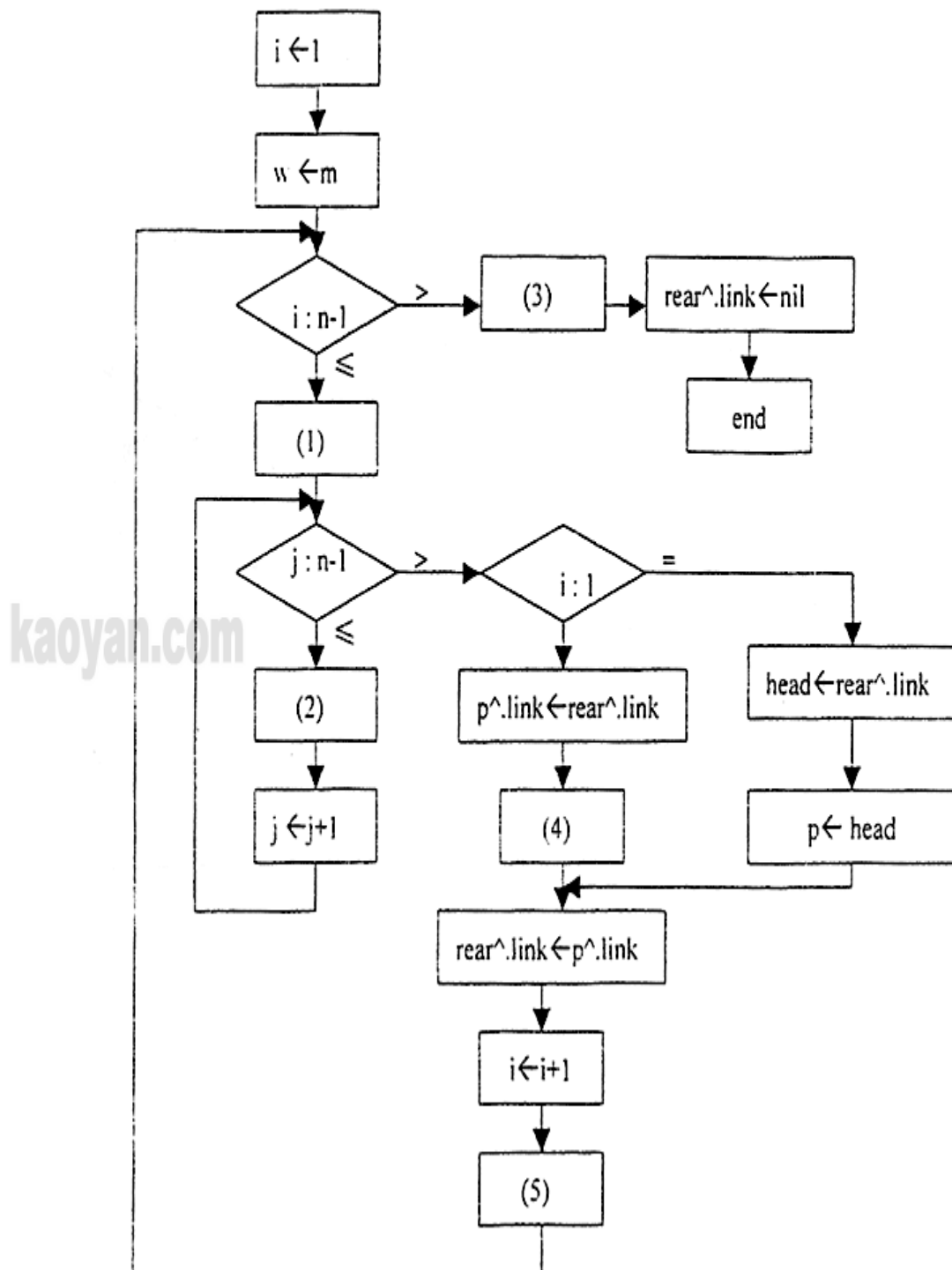
```
    end
```

3. 约瑟夫问题叙述如下：有 1 至  $N$  编号的  $N$  个人按顺时针方向围坐一圈，每人持有一个密码（正整数），一开始以正整数  $m$  作为报数上限值，从第一个人开始顺时针方向自 1 开始顺序报数，报到  $m$  时停止报数，报  $m$  的人出列，将他的密码作为新的报数上限值，从他的顺时针方向上的下一个人开始重新报数，如此下去，直至所有的人全部出列为止，要求产生记录出列顺序的表。如  $N=7$ ，每个人的密码依次是：3, 1, 7, 2, 4, 8, 4,  $m$  的值为 20，则出列顺序为 6, 1, 4, 7, 2, 3, 5。所有人用一个循环单链表表示，表中每个结点代表一个人，按出列次序依次将结点从循环单链表中删除，并按顺序存放在一个单链表中，链表的每个结点包括三个字段：code 代表密码，no 代表人的编号，link 是指向下一个结点的指针。初始地址 rear 指向循环单链表的表尾，最后指针 head 指向表示出列序列的单链表的表头。（10 分）

算法 (a)

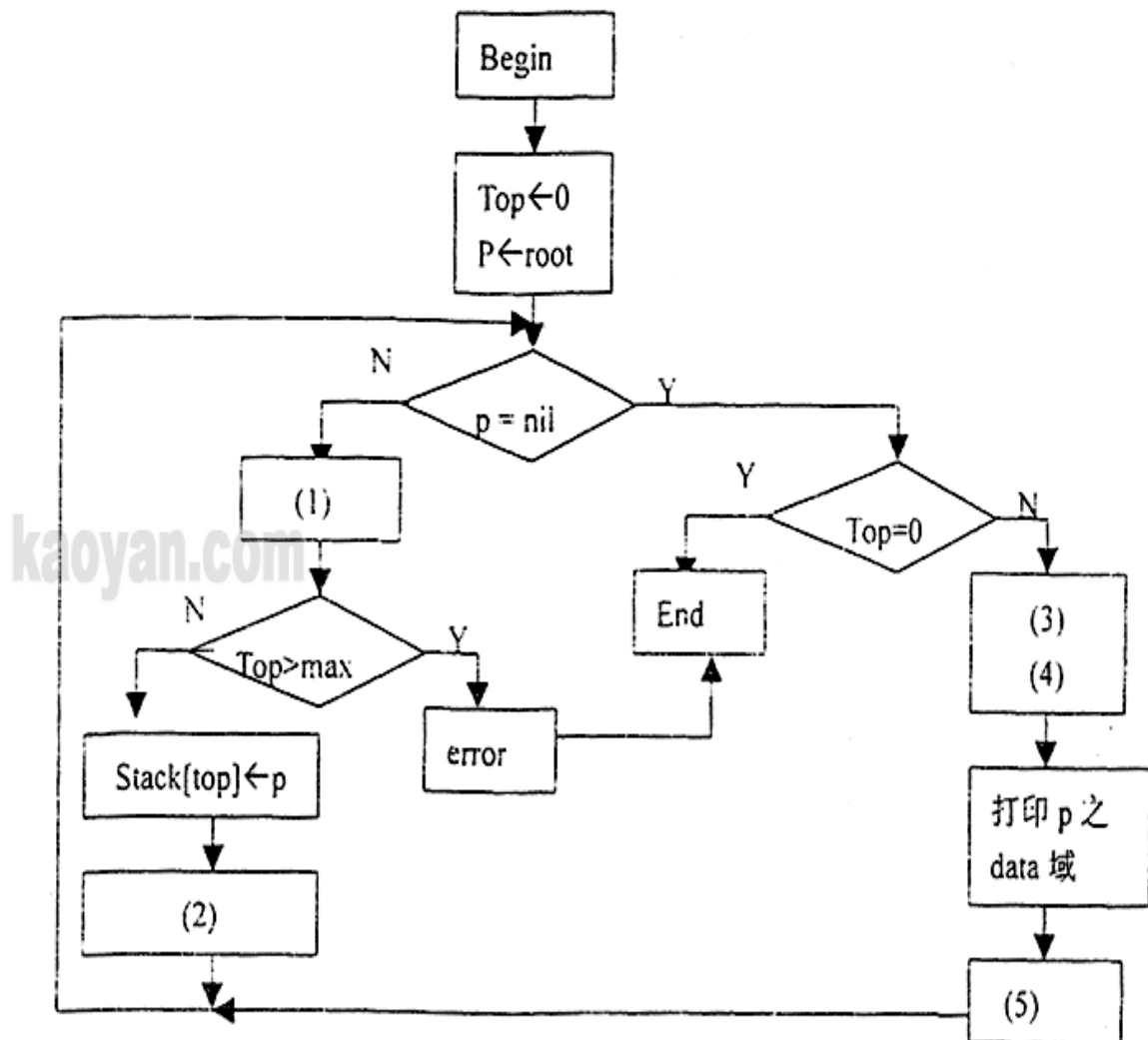


算法 (b)





4. 二叉树用二叉链表表示, 下面是前序遍历二叉树 root 的非递归算法, p 为当前结点, s 为辅助栈, s: satck[1..max] of pointerb, pointerb 即为二叉链表的结点类型。(10分)



## 二、请编写完整的程序。(共 3 2 分)

1. 请打印输出不大于  $M$  ( $M$  为  $\geq 3$  的正整数) 的所有史密斯数。史密斯数是指满足下列条件的可分解的整数：其所有数位上的数字和等于其全部素数因子之和。如  $9975 = 3 \times 5 \times 5 \times 7 \times 17$ ,  $9 + 9 + 7 + 5 = 3 + 5 + 5 + 17 = 30$ 。(14 分)
2. 设有顺序放置的  $n$  个桶，每个桶中装有一粒砾石，每粒砾石的颜色是红、白、蓝之一。要求重新安排这些砾石，使得所有红色砾石在前，所有白色砾石居中，所有蓝色砾石居后。重新安排时，对每粒砾石的颜色只能察看一次，并且只允许用交换操作来调整砾石的位置。(18 分)

## 三、请用流程图或类高级语言 (pascal 或 c) 表示算法。

(共 3 2 分)

1. 一棵二叉树以二叉链表来表示，求其指定的某一层  $k$  ( $k > 1$ ) 上的叶子结点的个数。(18 分)
2. 写算法判别以邻接表方式存储的无向图中是否存在由顶点  $V_i$  到顶点  $V_j$  的路径 ( $i \neq j$ )。(14 分)

