

上海大学 2002 年攻读硕士学位研究生 入学考试试题

招生专业: 计算机系统结构
计算机软件与理论
计算机应用技术

考试科目: 程序设计与数据结构一

一、(8 分) 请写出应填入下列叙述中 内的正确答案。

排序有各种方法, 如插入排序、快速排序、堆排序等。设一数组中原有数据如下: 15, 13, 20, 18, 12, 60 下面是一组用不同排序方法进行一遍排序后的结果。

- A 排序的结果为: 12, 13, 15, 18, 20, 60
 B 排序的结果为: 13, 15, 18, 12, 20, 60
 C 排序的结果为: 13, 15, 20, 18, 12, 60
 D 排序的结果为: 12, 13, 20, 18, 15, 60

二、(12 分) 请写出应填入下列问题 1 和 2 的叙述中 内的正确答案, 并解答问题 3。

- (2 分) 组成循环链表的可利用空间表当附加了条件 A 和 B 时, 首次拟合法即为最佳拟合法。
- (2 分) 二进制地址为 011011110000, 大小为 $(4)_{10}$ 和 $(16)_{10}$ 块的伙伴地址分别为: C 、 D 。
- (8 分) 假设利用边界标识法, 并以首次拟合策略分配, 已知在某个时刻可利用空间表的状态如图 1 所示:

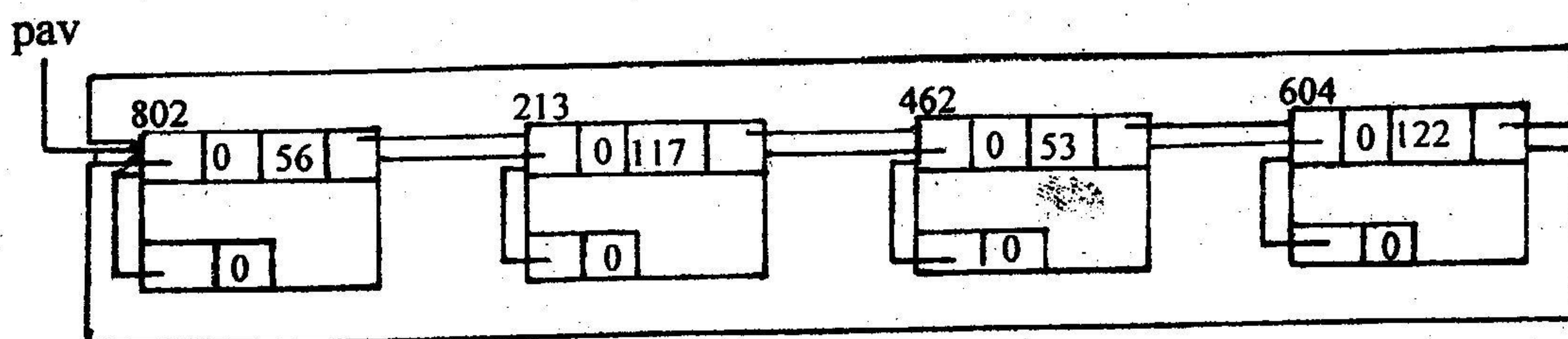


图 1 可利用空间表的状态图

(注: 存储块头部 size 域的值和申请分配的存储量均包括头部和尾部的存储空间。)

请画出:

- 1) 当系统回收一个起始地址为 559, 大小为 45 的空闲块之后的链表状态;

第 2 页 (共 10 页)

2) 系统继而在接受存储块大小为 100 的请求之后, 又回收了一个起始地址 515, 大小为 44 的空闲块之后的链表状态。

三、(10 分) 请写出应填入下列叙述中____内的正确答案。

某一工程作业的网络图如图 2 所示。其中箭头表示作业, 箭头边的数字表示完作业所需的天数。箭头前后的圆圈表示事件, 圆圈中的数字是事件的编号。用事件号的序列 (例如 0-2-7-9-11) 表示进行作业的路径。

完成此工程的关键路径是 A 完成此工程所需的最少天数为 B 天, 工程中具有最大充裕天数的的事件是 C, 充裕天数是 D。关键路径上的事件的裕天数是 E。

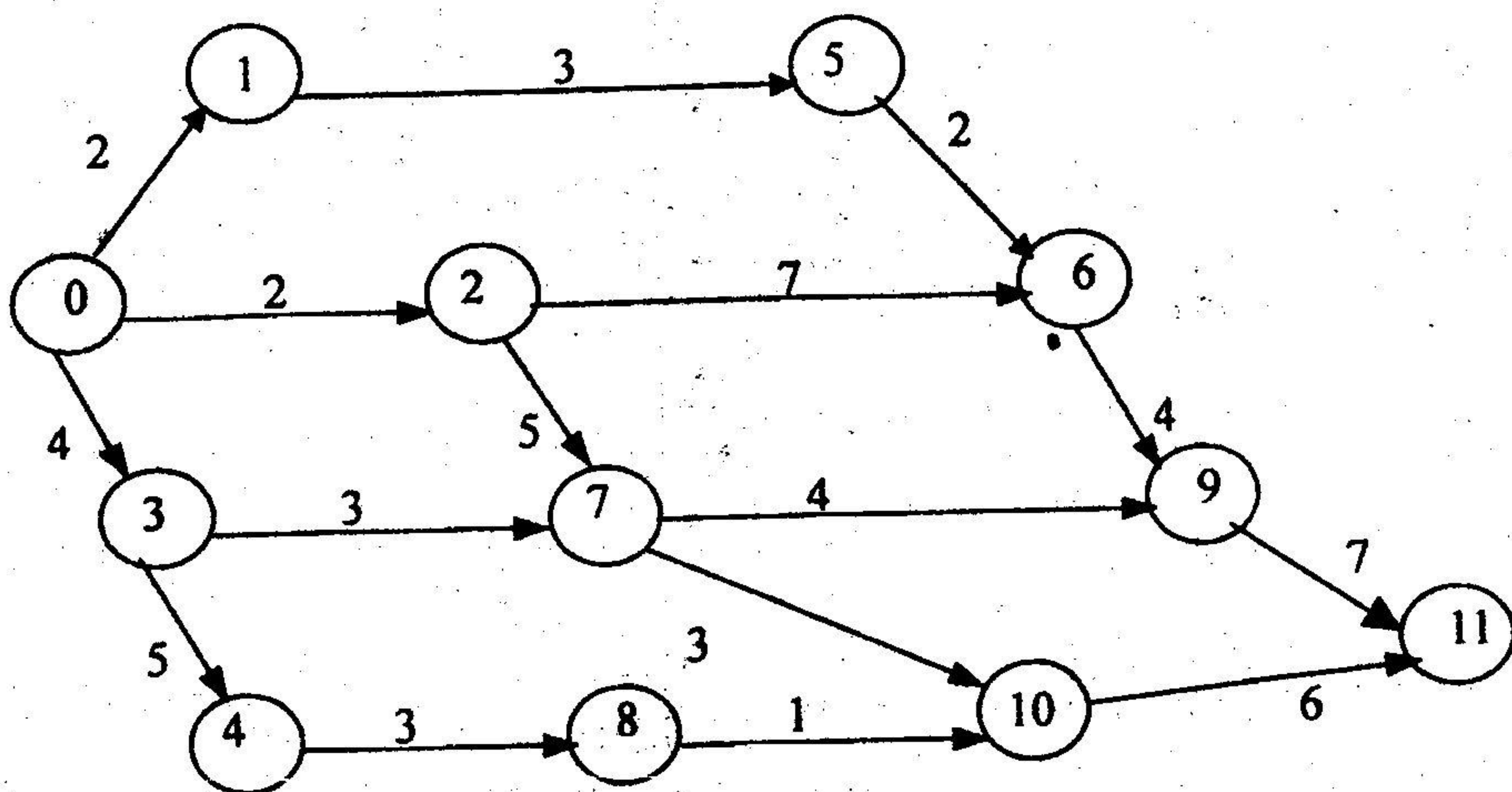


图 2. 工程作业的网络图

四、(10 分) 按下列要求构造二叉树。

1. (4 分) 已知一棵二叉树的前序序列为 ABDEGCF, 中序序列为 DBGEACF, 请写出该二叉树后序序列。
2. (6 分) 设数据集合 $d = \{1, 12, 5, 8, 3, 10, 7, 13, 9\}$,
 - 1) 依次取 d 中各数据构造一棵平衡二叉树 AVL;
 - 2) 如何依据此平衡二叉树 AVL 得到 d 的一个有序序列;
 - 3) 画出在 AVL 中删除 “5” 后的平衡二叉树。

五、(10 分) 阅读下列说明和流程图(图 3), 在该图中①~⑤处填入合适的字句, 使成为完整的流程图。

[说明] 流程图 (图 3) 用来求满足下列条件的最小自然数 N : 对于两个给定的素数 Y , 从 N 开始的每个自然数都能表示成 X 的倍数和 Y 的倍数之和, 即表示 $m_1 \cdot X + m_2 \cdot Y$, 其中 m_1, m_2 均为自然数。

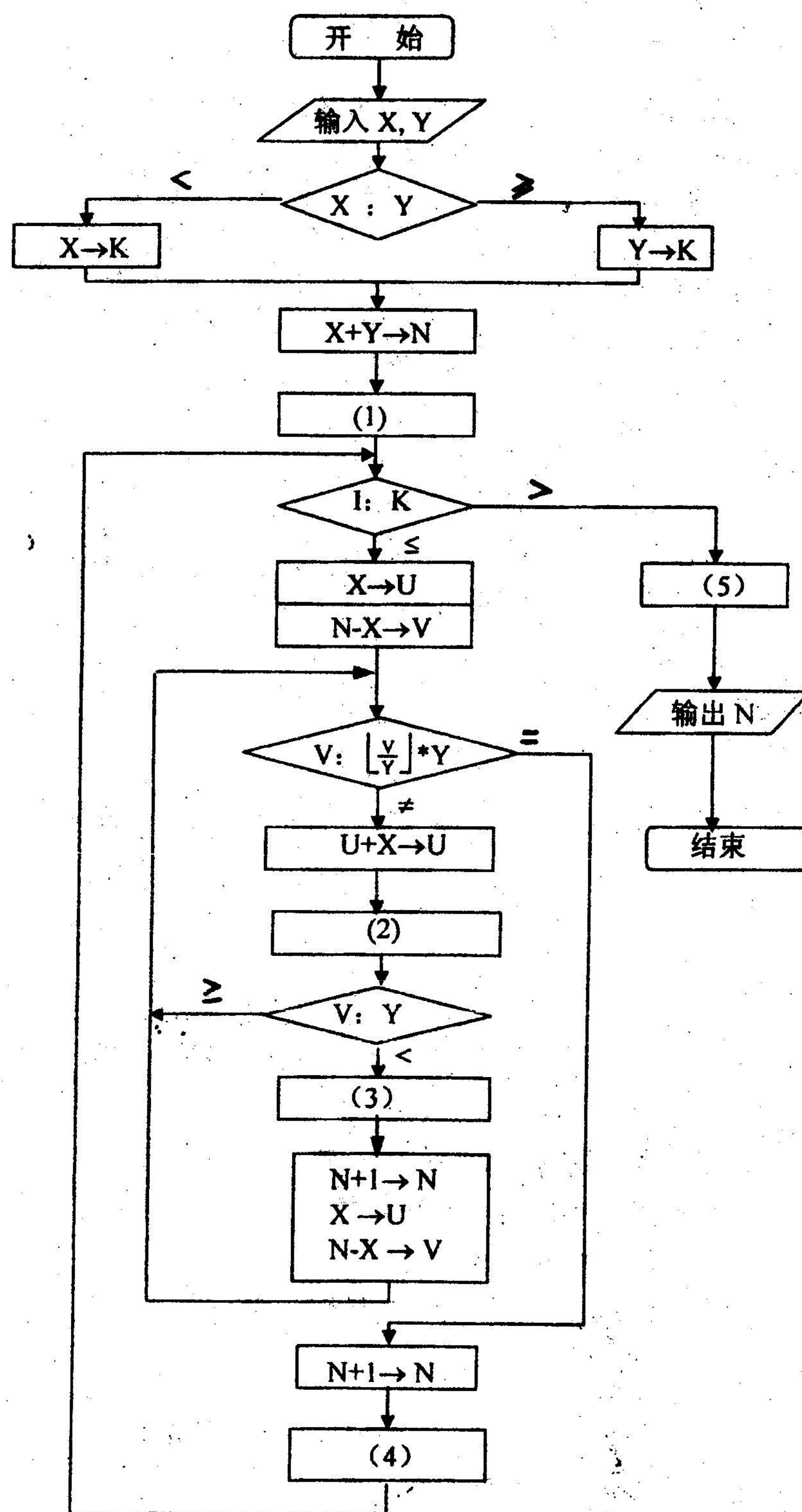


图 3. 流程图

第 4 页 (共 10 页)

寻找 N 的过程如下：首先取 X 和 Y 中的较小者为 K ，然后从 $X+Y$ 开始逐个检查每个自然数，看它是否能表示成 X 的倍数和 Y 的倍数之和，一旦找到 K 个连续的自然数都符合该条件，则这 K 个数中的第一个即为所求的 N 的值。

对每个数的检查方法是：将该数看作 U 、 V 两数之和（即 $N=U+V$ ），取 U 为 X 的倍数，然后检查 V 是否能被 Y 整除。

本题假设输入的 X 、 Y 是两个素数。图中 $[W]$ 表示不超过 W 的最大整数。

六、(30 分) 完善下列程序，每小题在 PASCAL 语言(a)和 C 语言(b)中任选一题。若都做，按(a)记分，每小题 10 分。

1. 下面的程序将数列 $1, 2, 3, \dots, n \times n$ ，依次按蛇型方式存放在二维数组 $A[1..n, 1..n]$ 中。即：

1	2	6	7	...
3	5	8	14	...
4	9	13
10	12
11
...

(a) 算法的 PASCAL 语言程序描述：

```

PROGRAM ex1(INPUT,OUTPUT);
CONST NMAX=10;
VAR
  a: ARRAY [1..NMAX, 1..NMAX] OF integer;
  i, j, n, k, p, q, m: integer;
BEGIN
  readln(n); m:=1;
  FOR k:=1 TO ① DO
  BEGIN
    IF k<n THEN q:=k
      ELSE ②
    FOR p:=1 TO q DO
    BEGIN
      IF ③ THEN
      BEGIN
        i:=q-p+1; j:=p
      END
    END
  END

```



```

ELSE
BEGIN
    i:=p; j:=q-p+1
END;
IF ④ THEN
BEGIN
    i:=i+n-q; j:=j+n-q
END;
a[i, j]:=m;
    ⑤
END
END;
FOR i:=1 TO n DO
BEGIN
    FOR j:=1 TO n DO
        write(a[i][j]:4); writeln;
    END
END.

```

(b) 算法的 C 语言程序描述:

```

#define NMAX 10
#include "stdio.h"
main()
{
    int i, j, n, k, p, q, m;
    int a[NMAX][NMAX];
    scanf ("%d", &n);
    m=1;
    for (k=1; ①; k++)
    {
        if (k<n) q=k;
        else ②
        for (p=1; p<=q; p++)
        {
            if (③)
            { i=q-p+1; j=p;}
            else { i=p; j=q-p+1;}
        }
    }
}

```



```

if ( ④ )
    { i=i+n-q; j=j+n-q; }
    a[i][j]=m; ⑤
}
for (i=1; i<=n; i++)
{
    for (j=1; j<=n; j++)
        printf("%4d", a[i][j]); printf("\n");
}
}
}

```

2. 下面是采用单链表作存储结构实现选择排序(升序) 的程序段。

(a) 算法的 PASCAL 语言过程描述:

TYPE

linklist=^node;

node=RECORD

key:integer;

next:linklist

END;

PROCEDURE selectsort(VAR h:linklist);

VAR p,q,r,s,t:linklist;

BEGIN

t:=nil;

WHILE ① DO

BEGIN

p:=h; q:=nil;

s:=h; r:=nil;

WHILE (②) DO

BEGIN

IF (③) THEN

BEGIN

s:=p; r:=q

END;

q:=p;

p:=p^.next

END;


```

    IF (s=h) THEN    h:=h^.next
        ELSE      ④ ;
    ⑤ ;
    t:=s;
END;
h:=t;
END;

```

(b) 算法的 C 语言过程描述:

```
typedef struct node
```

```
{
```

```
    int key;
```

```
    struct node *next;
```

```
} *LINKLIST;
```

```
LINKLIST selectsort(h)
```

```
    LINKLIST h;
```

```
{
```

```
    LINKLIST p, q, r, s, t;
```

```
    t=NULL;
```

```
    while ( ① )
```

```
    {
```

```
        p=h; q=NULL;
```

```
        s=h; r=NULL;
```

```
        while ( ② )
```

```
        {
```

```
            if ( ③ )
```

```
            { s=p; r=q; }
```

```
            q=p; p=p->next;
```

```
        }
```

```
        if (s==h) h=h->next;
```

```
        else ④ ;
```

```
        ⑤ ;
```

```
        t=s;
```

```
    }
```

```
    h=t ;
```

```
}
```


第 8 页 (共 10 页)

3. 下面是一个将广义表逆置的过程。例如原来广义表为((a, b), c, (d, e)), 经逆置后为((e, d), c, (b, a))。

(a) 算法的 PASCAL 语言过程描述:

TYPE tagtype=0..1;

glist=^gnode;

gnode=RECORD

CASE tag:tagtype OF

0:(data:char);

1:(hp,tp:glist);

END;

PROCEDURE reverse(p:glist;VAR q:glist);

VAR h,t,s:glist;

BEGIN

IF (p=nil) THEN q:=nil

ELSE BEGIN

IF (①) THEN

BEGIN

new(q); q^.tag:=0;

q^.data:=p^.data;

END

ELSE BEGIN

②;

IF (③) THEN

BEGIN

reverse(p^.tp,t); s:=t;

WHILE s^.tp<>nil DO s:=s^.tp;

new(s^.tp);s:=s^.tp;

s^.tag:=1; s^.tp:=nil; s^.hp:=h;

④;

END

ELSE BEGIN

new(q); ⑤ ;

q^.tp:=nil;q^.tag:=1

END

END

END

END;

(b) 算法的 C 语言过程描述:

```
typedef struct glistnode
{
    int tag;
    struct glistnode *next;
    union
    {
        char data;
        struct { struct glistnode *hp,*tp;} ptr;
    } val;
} *glist,gnode;

glist reverse(p)
glist p;
{ glist q,h,t,s;
  if ( p==NULL) q=NULL;
  else
  {
    if ( ① )
    {
      q=(glist)malloc(sizeof(gnode)); q->tag=0;
      q->val.data=p->val.data;
    }
    else
    {
      ② ;
      if ( ③ )
      {
        t=reverse(p->val.ptr.tp); s=t;
        while (s->val.ptr.tp!=NULL) s=s->val.ptr.tp;
        s->val.ptr.tp=(glist)malloc(sizeof(gnode));
        s=s->val.ptr.tp; s->tag=1; s->val.ptr.tp=NULL;
        s->val.ptr.hp=h;
        ④ ;
      }
      else
      {
        q=(glist)malloc(sizeof(gnode)); q->tag=1;
        q->val.ptr.tp=NULL; ⑤ ;
      }
    }
  }
  return(q);
}
```


七、(20 分, 每小题 10 分) 请用类 C 或类 PASCAL 编写算法。

1. 请编写在中序全线索二叉树 T 中的结点 p 下插入一棵根为 x 的中序全线索二叉树的算法。如果 p 左右孩子都存在, 则插入失败并返回 false; 如果 p 没有左孩子, 则 x 作为 p 的左孩子插入; 否则 x 作为 p 的右孩子插入。插入完成后要求二叉树保持中序全线索并返回 true。

2. 键树, 又称数字查找树。它是一棵度 ≥ 2 的树, 树中的每个结点中不是包含一个或几个关键字, 而是只含有组成关键字的符号。编写一个在键 (Trie) 树 T 上查找关键字等于给定值 key 的记录的算法。若查找成功, 返回指向该记录的指针; 否则返回空指针。