



算法伪代码 (Pseudo Code)

高晓枫 (Xiaofeng Gao)

Department of Computer Science
Shanghai Jiao Tong Univ.

程序的概念

- ❖ **程序 (Program)**: 可以被计算机处理的指令序列。
 - 程序是为完成一项任务、由汇编语言或高级语言编写的代码的集合。
- ❖ **程序设计**: 根据所提出的任务, 用某种程序设计语言编制一个能正确完成该任务的计算机程序。

程序=数据结构+算法

——Nikiklaus Wirth

(沃思, 著名计算机科学家)

程序的性质

- ❖ 程序指令**顺序**执行。
 - ❖ 程序执行结束时应有一个**结果**。
 - ❖ 程序要对某些**对象**进行操作。
 - ❖ 有的程序要加入对操作对象的**说明**。
 - ❖ 有时指令要求执行者做出**判断**。
 - ❖ 一条或一组指令可能需要**执行多次**。
- ❖ 程序流程：
问题定义→算法设计→程序编制→调试测试及资料编制

算法特性

❖ 程序设计离不开算法，算法指导程序设计，是程序的灵魂。算法是精确定义的一系列规则。

❖ 算法特性：

- (1) 有效性。
- (2) 确定性。
- (3) 有穷性。
- (4) 有零个或多个输入。
- (5) 有一个或多个输出。

有效性指算法所规定的操作都应当能够有效执行。

确定性一是:所描述的操作应当具有明确的意义，不应当有歧义性。

二是:操作序列只有一个初始动作，序列中每一动作仅有一个后继动作；

有穷性指算法所规定的操作序列必须在允许的时间内结束。

算法伪代码 (Pseudocode)

- ❖ **伪代码 (Pseudocode)**: 一种算法描述语言, 使之可用任何编程语言 (Pascal, C, Java, etc) 实现。
- ❖ **伪代码特点**:
 - 结构清晰、代码简单、可读性好。
 - 类似自然语言, 介于自然语言与编程语言之间。
 - 以编程语言的书写形式指明算法职能。
 - 不用拘泥于具体实现。
- ❖ 可描述整个算法运行过程的结构, 表述算法功能 (而不是其语言实现)。用于技术文档和科学出版物中表示算法, 也用于在软件开发的实际编码过程之前表达程序的逻辑。

算法名称与初始化

❖ 无参数表示

Algorithm 6: Gender

Input: *person*

Output: *person's* gender

❖ 带参数表示

Algorithm 7: $\max(a, b)$

Input: a, b

Output: the max of a and b

基本控制语句

❖ 赋值语句：中间变量定义等，用 \leftarrow 表示。

$x \leftarrow y;$ // 将 y 的值赋值给 x
 $x \leftarrow 0;$
 $x \leftarrow x - 1;$

有时也用
:=或者=
表示赋值

❖ 线性表述（顺序结构，SEQUENCE），下一个命令会在当前命令结束时开始。

1 $x \leftarrow 1;$
2 $x \leftarrow 10;$
3 $y \leftarrow x - 8;$
4 $x \leftarrow x - y;$
5 **return** x

$x = ?$

流程图表示法



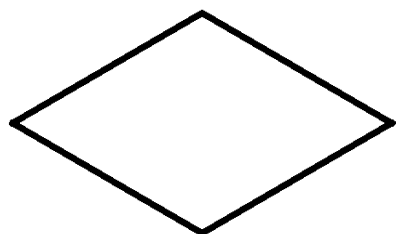
开始/结束



输入/输出



语句处理



条件判断



需进一步展开



连接符

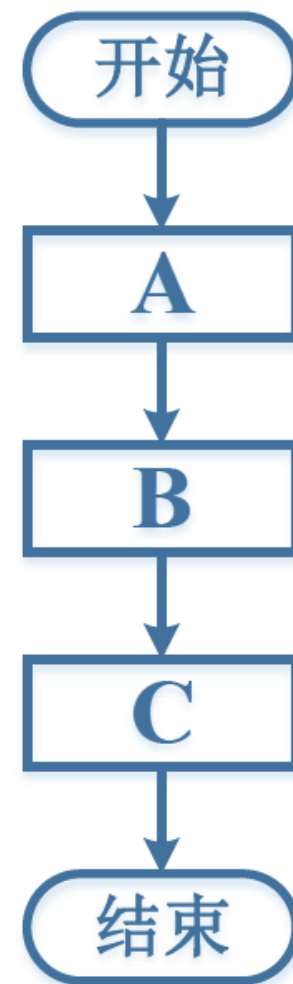


流程线

流程图表示法常用图例

顺序结构的流程图表示

- ❖ 顺序结构是一种简单的线性结构，根据流程线所示的方向，各矩形框按顺序执行。
- ❖ 如：右图语句的执行顺序为：
 $A \rightarrow B \rightarrow C$



选择控制语句

- ❖ IF-THEN-ELSE: 基本选择语句, 选择结构, 其中 else 可选, 还可插入多项 ELSEIF

```
if condition then  
| sequence 1;  
end
```

```
if condition then  
| sequence 1;  
else  
| sequence 2;  
end
```

```
if condition 1 then  
| sequence 1;  
else if condition 2 then  
| sequence 2;  
else if condition 3 then  
| sequence 3;  
else  
| sequence 4;  
end
```

选择控制语句示例

❖ 整除标记

Algorithm 1: $div(x, y)$

Input: $x \geq 0, y > 0$

Output: $sign$

```
1 if  $rm(x, y) = 0$  then
2   |  $sign \leftarrow 1;$ 
3 else
4   |  $sign \leftarrow 0;$ 
5 end
6 return  $sign;$ 
```

❖ 作业批改

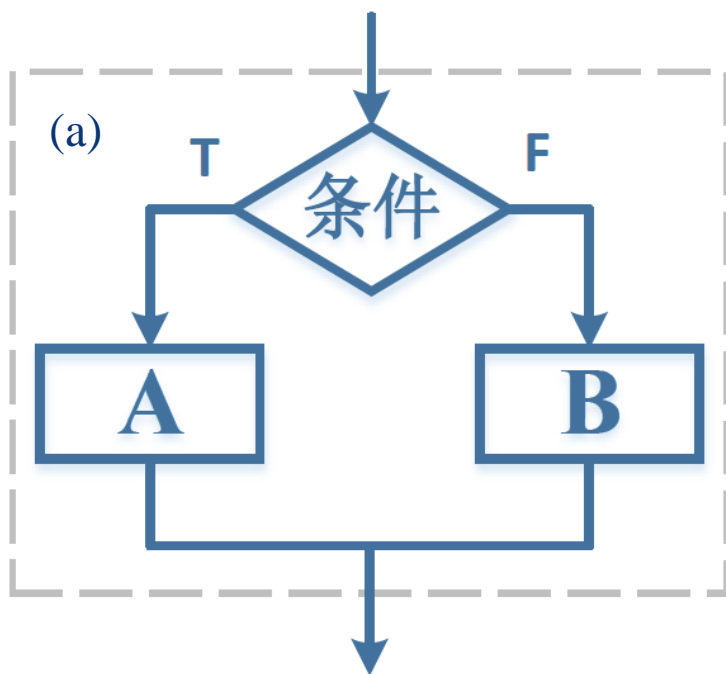
Algorithm 2: LetterGrade($score$)

Input: $score$

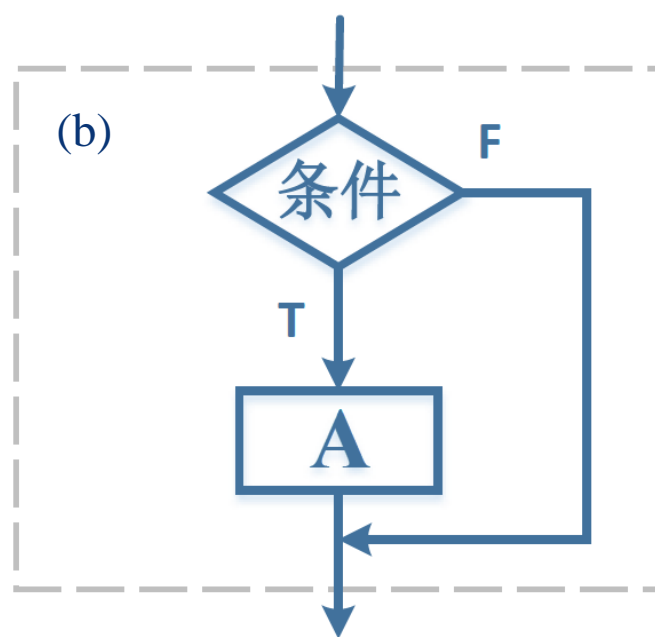
Output: Letter Grade

```
1 if  $score \geq 90$  then
2   | output  $A;$ 
3 else if  $80 \leq score < 90$  then
4   | output  $B;$ 
5 else
6   | output  $P;$ 
7 end
```

顺序结构的流程图表示



(a)的执行顺序为：先判断条件，当条件为真时，执行A，否则执行B。



(b)的执行顺序为：先判断条件，当条件为真时，执行A，否则什么也不执行。

复合条件控制语句

❖ 格式:

〈简单条件〉〈逻辑连接符〉〈简单条件〉

❖ 例:

- 合取: $(a < b + c) \wedge (b < a + c) \wedge (c < a + b)$
- 析取: $(x = 5) \vee (x = 8)$
- 有时也用 **and**, **or** 来表示

While型循环控制语句

❖ WHILE: 循环语句, 在初始阶段判断条件。当条件为真时执行循环, 条件为假时跳出循环。

```
while condition do  
| sequence;  
end
```

❖ REPEAT-UNTIL: 循环语句, 在循环尾部进行控制。条件为真时跳出。

```
repeat  
| sequence;  
until condition;
```

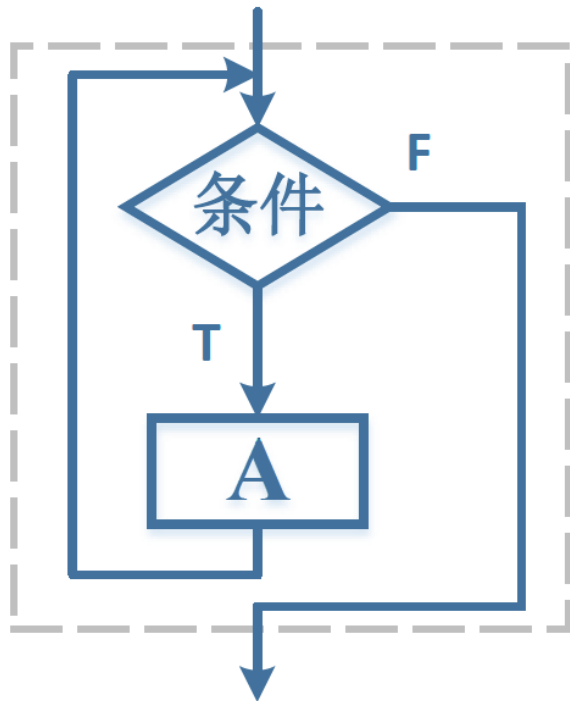
❖ 有时也做DO-WHILE

```
do  
| sequence;  
while condition;
```

While型结构的流程图表示

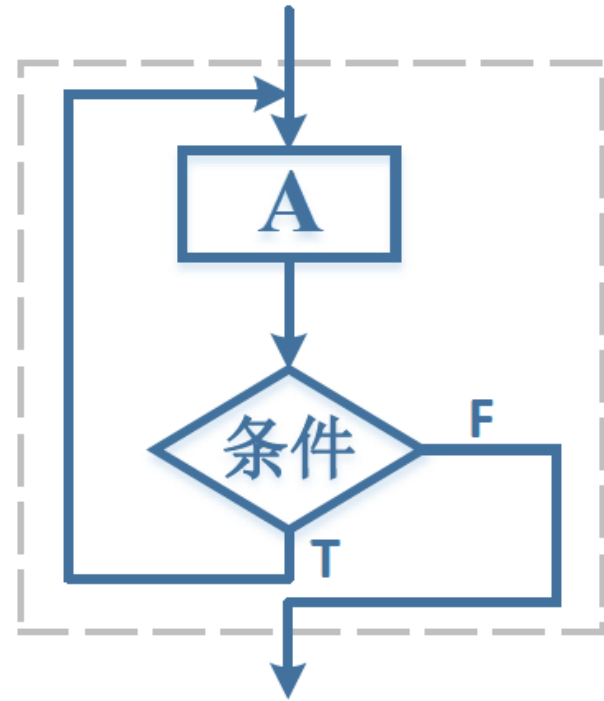
①while型（当型）循环

重复判断条件、如果条件为真则执行A，一旦条件为假，则跳出循环。



②do-while（直到型）型循环

先执行A，再判断条件，若条件为真则重复执行A，若条件为假，则跳出循环。



A被称为循环体，条件被称为循环控制条件。

While型循环控制语句示例

❖ Remainder

Algorithm 1: $rm(x, y)$

Input: $x > 0, y > 0$

Output: x

```
1 while  $x \geq y$  do
2   |  $x - = y;$ 
3 end
4 output  $x;$ 
```

❖ GCD

Algorithm 2: $\text{GCD}(a, b)$

Input: $a, b \in \mathbb{N}$

Output: Greatest common
divisor of a, b

```
1 repeat
2   |  $gcd \leftarrow a \bmod b;$ 
3   |  $a \leftarrow b;$ 
4   |  $b \leftarrow gcd;$ 
5 until  $gcd = 0;$ 
6 output  $a;$ 
```

For型循环控制语句

❖ FOR: 循环次数受限的循环语句。(counting loop)

```
for iteration bound do  
  | sequence;  
end
```

```
foreach  $x \in A$  do  
  | sequence;  
end
```

```
forall the employee do  
  | sequence;  
end
```

Example

Algorithm 4: Sum(n)

Input: $n \in \mathbb{N}$

Output: The sum from 1 to n

```
1  $sum \leftarrow 0$ ;  
2 for  $temp = 0$  to  $n$  do  
3   |  $sum \leftarrow sum + temp$ ;  
4 end  
5 output  $sum$ ;
```

多分支选择控制语句

❖ SWITCH-CASE: 多分支选择语句 (default optional)

```
switch the value of T do
| case value 1
|   sequence 1;
| case value 2
|   sequence 2;
| otherwise
|   sequence 3;
endsw
endsw
```

Algorithm 6: Gender

Input: *person*

Output: *person's* gender

```
1 switch person do
2   | case person.gender = male
3     | output Male;
4   | case person.gender = female
5     | output Female;
6   | otherwise
7     | output Unknown;
8   | endsw
9 endsw
```

嵌套叠加

- ❖ 基本控制语句可以嵌套叠加，由缩进严格控制。
- ❖ 判断Judge的功能。
 - 这个算法能够判断一个自然数是否是质数。

Algorithm 1: Judge(a, b)

Input: $x \in \mathbb{N}$

Output: $\{True, False\}$

```
1 if  $x < 2$  then
2   | return False;
3 for  $2 \leq i \leq \sqrt{x}$  do
4   | if  $x \% i = 0$  then
5     | return False;
6 return True;
```

子函数引入(subfunction)

❖ 用调用函数名形式引入子函数

Algorithm 1: $\text{sort}(x, y)$

Input: $x, y \in \mathbb{N}$

Output: \emptyset

```
1 if  $x < y$  then  
2   |    $tmp \leftarrow x$ ;  
3   |    $x \leftarrow y$ ;  
4   |    $y \leftarrow tmp$ ;  
5 end
```

Algorithm 2: $\text{gcd}(x, y)$

Input: $x, y \in \mathbb{N}^+$

Output: $ret \in \mathbb{N}$

```
1  $\text{sort}(x, y)$ ;  
2 while  $x \% y \neq 0$  do  
3   |    $x \leftarrow x \% y$  ;  
4   |    $\text{sort}(x, y)$ ;  
5 end  
6  $ret \leftarrow y$ ;  
7 return  $ret$ ;
```

常用表达

- ❖ `i++;` `++i;`
- ❖ `x /= 2;`
- ❖ `x := 0;`
- ❖ `y == z;`
- ❖ `For (i=1 to 10);`
- ❖ `A&&B;`
- ❖ `A || B;`
- ❖ `BREAK;`
- ❖ `CONTINUE;`
- ❖ 将变量*i*的值加1
- ❖ `x=x/2`
- ❖ `x`赋值为0
- ❖ 判断*y*与*z*是否相等
- ❖ For循环控制语句
- ❖ A和B
- ❖ A或B
- ❖ 停止循环跳出
- ❖ 停止当前循环进入下一轮

Break和Continue

Algorithm 1: Break and Continue

Input: *null*

Output: *tmp*

```
1 tmp ← 0;
2 for i = 1 to 10 do
3   if i mod 2 = 0 then
4     Continue;
5   else
6     tmp ← tmp + 1;
7     if tmp = 5 then
8       Break;
9   Print "iteration=i, temp=tmp";
10 output tmp;
```

❖ 执行顺序

- *i*=1, Print
- *i*=2, 跳过
- *i*=3, Print
- *i*=4, 跳过
- *i*=5, Break

范例

❖ 例：找出三个数中最大的那个数。

- 分析：三个数值必须用三个变量来保存，假定这三个变量分别为a、b、c，另外，还需要一个变量max来装最小的那个数。
- 先比较a和b的值，把数值大的放入max中，再将max与c比较，又把数值大的放入max中。
- 经过两次比较，max中已存放的是a，b，c三个数中最大的数，把max的值输出就是所需结果。

伪码(1)

Algorithm 1: Maximum Number

Input: a, b, c

Output: The maximum value of a, b, c

```
1 if  $a > b$  then
2   |    $max = a;$ 
3 else
4   |    $max = b;$ 
5 end
6 if  $c > max$  then
7   |    $max = c;$ 
8 end
9 output  $max;$ 
```

伪码(2)

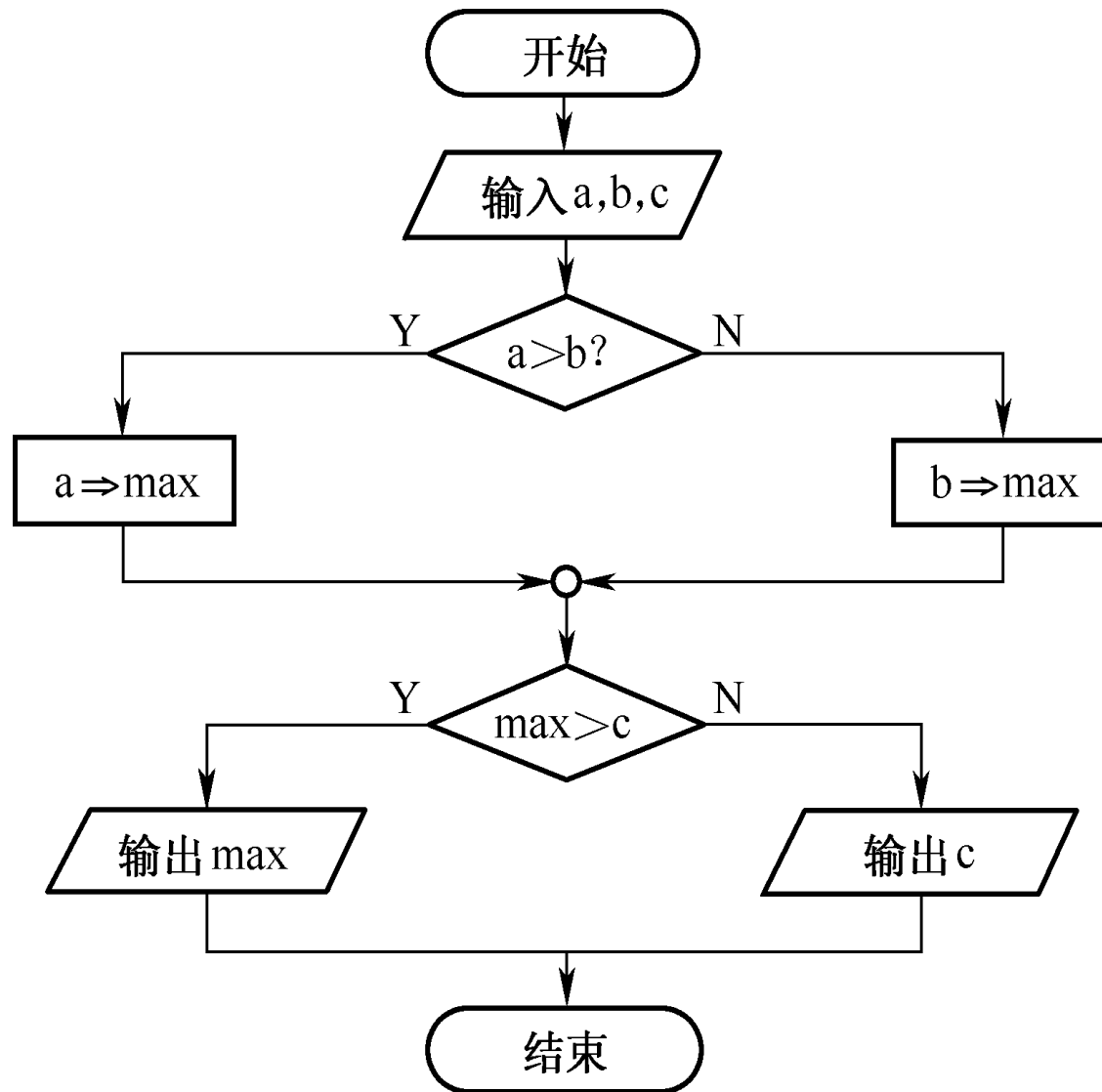
Algorithm 1: Maximum Number

Input: a, b, c

Output: The maximum value of a, b, c

```
1 if  $a > b$  then
2   |    $max = a;$ 
3 else
4   |    $max = b;$ 
5 if  $c > max$  then
6   |    $max = c;$ 
7 output  $max;$ 
```

流程图表示





PROBLEM PSEUDO CODE?

Thank You !

Xiaofeng Gao