

# NOIP 普及组初赛单项选择复习资料

## 计算机结构与组成原理

### 一、计算机发展及应用

#### 1、第一台电子计算机的诞生：ENIAC

1946 年，世界上第一台数字式电子计算机是由美国宾夕法尼亚大学的物理学家约翰·莫克利（John Mauchly）和工程师普雷斯伯·埃克特（J. hesper. Eckert）领导研制的取名为 ENIAC（Elecotmnic Nurnerical Integrator And Calculator）的计算机。

#### 2、第一台具有存储程序功能的计算机：EDVAC。

1903 年，冯·诺伊曼（Neumann, John von）出生于匈牙利的布达佩斯。冯·诺依曼依据存储程序的工作原理设计。

运算器、控制器、存储器、输入设备和输出设备这五部分组成，同 ENIAC 相比，EDVAC 方案有两个重大改进：（1）：采用了二进制；（2）：提出了“存储程序”。

#### 3、图灵机和图灵奖

艾伦·麦席森·图灵（Alan Mathison Turing, 1912 年 6 月 23 日 - 1954 年 6 月 7 日），英国数学家。

图灵机由三部分组成，包括一条带子、一个读写头和一个控制装置。

图灵对于人工智能的发展有诸多贡献，例如：图灵曾写过一篇名为《机器会思考吗？》（Can Machine Think?）的论文，其中提出了一种用于判定机器是否具有智能的试验方法，即图灵试验。

图灵奖是美国计算机协会于 1966 年设立的，又叫“A.M. 图灵奖”，专门奖励那些对计算机事业作出重要贡献的个人。其名称取自计算机科学的先驱、英国科学家艾伦·图灵，这个奖设立目的之一是纪念这位科学家。

#### 4、世界上第一位软件工程师

英国著名诗人拜伦的女儿 Ada Lovelace(爱达). 由于她在程序设计上的开创性工作，Ada Lovelace 被称为世界上“第一位程序员”。“世界上第一位软件工程师”。

#### 5、微型计算机的问世

第四代 1972——至今 超大规模集成电路的微星计算机个人 PC 应用到了各个领域。

### 二、硬件系统的组成：

#### 1、冯·诺伊曼体系

其思想是，在计算机中设置存储器，将符号化的计算步骤存放在存储器中，然后依次取出存储的内容，由一个被称之为控制器的部件进行译码，译码结果在一个被称为运算器的部件中进行计算，从而实现计算机工作的自动化（运算器和控制器统称为CPU）。

五个基本部分组成：（1）运算器，（2）控制器，（3）存储器，（4）输入设备，（5）输出设备

计算机的整个工作过程及基本硬件结构如图2-8所示：

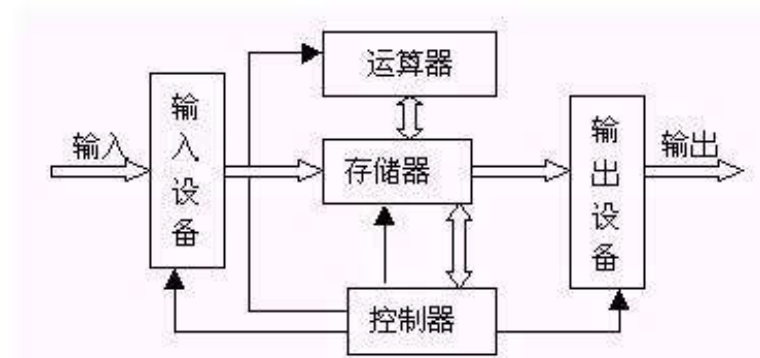
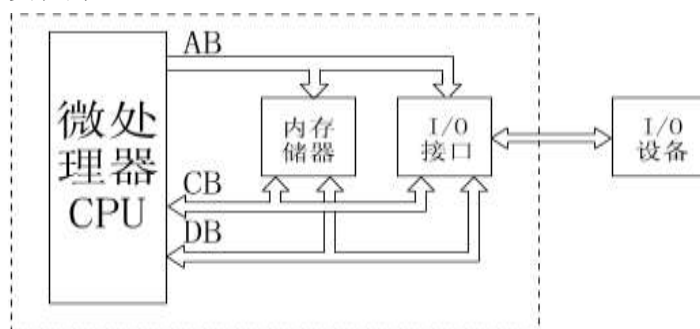


图2-8 计算机系统的基本硬件组成及工作原理

**存储器简单分类：**寄存器和高速缓存；RAM和ROM；软盘和硬盘。（内部、外部存储器）

## 2、计算机的三总线结构

总线是一组导线、是公共通路，微型计算机中各个组成部件之间的信息传输都是通过它们来实现的



地址总线(AB)是单向总线，用以传送 CPU 向外设或存储器发出的地址信息。

数据总线(DB)是双向总线，用以 CPU 与内存或接口之间传输数据信息。

控制总线(CB)是双向总线，有的作为输出，有的作为输入，用以 CPU 与内存或 I/O 接口之间传送控制信息。

分别传送地址信号、数据信号和控制信号。

# 软件系统

## 1、系统软件：

(1) 操作系统软件：

DOS, OS/2 , Windows 9x, Windows 2000, Windows XP, Windows Vista, Win7  
Netware, Windows NT, Windows Server 200x, Unix, Linux, iOS4, Android 3.0

(2) 文件的后缀名：

bat、com、exe、sys、tmp、zip、……

doc、xls、txt、htm、……

gif、jpg、wav、avi、mp3、swf……

(3) 计算机语言：

机器语言，汇编语言；解释性语言和编译性语言。

高级语言：Logo, Basic, Pascal, c, c++, Visual Basic, Java, Go等。

## 2、应用软件：

Wps, Office (Word, PowerPoint, Excel), 3dmax, flash, photoshop等。

### 3、面向对象编程语言

面向对象语言 (Object-Oriented Language) 是一类以对象作为基本程序结构单位的程序设计语言，它之前呢？是面向过程。而现在呢？面向切面 (AOP)。

一种是纯面向对象语言，如Smalltalk、EIFFEL等。

混合型面向对象语言，即在过程式语言及其它语言中加入类、继承等成分，如C++、Objective-C等。Visual Basic, Java

面向切面 (AOP) 与面向对象 (OOP)，MM和OO的故事。

## 计算机中数字

数值信息在计算机内的表示方法就是用二进制数来表示。

一般说来，如果数制只采用R个基本符号，则称为基R数值，R称为数制的基数，而数制中每一固定位置对应的单位值称为权。

进制	基数R	基本符号
二进制	2	0, 1
八进制	8	0,1,2,3,4,5,6,7
十进制	10	0,1,2,3,4,5,6,7,8,9
十六进制	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F (对应十进制数的10—15, )

进位计数制的编码符合“逢 R 进位”的规则，各位的权是以 R 为底的幂，一个数可按权展开成为多项式。例如，一个十进制数 256.47 可按权展开为

$$256.47 = 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 4 \times 10^{-1} + 7 \times 10^{-2}$$

#### 1、R 进制转换为十进制

基数为 R 的数字，只要将各位数字与它的权相乘，其积相加，和数就是十进制数

例：  $3506.2_8$

$$= 6 \times 8^0 + 0 \times 8^1 + 5 \times 8^2 + 3 \times 8^3 + 2 \times 8^{-1}$$

$$= 1862.25$$

例：  $0.2A_{16}$

$$= 2 \times 16^{-1} + 10 \times 16^{-2}$$

$$= 0.1640625$$

#### 2、十进制转换为 R 进制

十进制整数转换成 R 进制的整数：除 R 取余法。

例：  $(89)_{10} = (1011001)_2$

2		89	
2		44	.....1
2		22	.....0
2		11	.....0
2		5	.....1
2		2	.....1
2		1	.....0
		0	.....1

十进制小数转换成 R 进制时：乘 R 取整。

例:  $(0.625)_{10} = (0.101)_2$

$$\begin{array}{r}
 0.625 \\
 \times 2 \\
 \hline
 1.25 \quad 1 \\
 \times 2 \\
 \hline
 0.5 \quad 0 \\
 \times 2 \\
 \hline
 1.0 \quad 1
 \end{array}$$

### 3、二、八、十六进制的相互转换

每位八进制数相当于三位二进制数，每位十六进制数相当于四位二进制数。在转换时，位组划分是以小数点为中心向左右两边延伸，中间的0不能省略，两头不够时可以补0。尤其是小数后末尾的0

例如：将  $1011010.1_2$  转换成八进制和十六进制数

$001\ 011\ 010.100 \quad 1011010.1_2 = 132.4_8$

1 3 2. 4

$0101\ 1010.1000 \quad 1011010.1_2 = 5A.8_{16}$

5 A. 8

例如：将十六进制数  $F7.28$  变为二进制数

$F \quad 7. \quad 2 \quad 8 \quad F7.28_{16} = 11110111.00101_2$

$1111\ 0111.0010\ 1000$

## 二、在计算机中带符号数的表示法

### 1、原码：

在用二进制原码表示的数中，符号位为0表示正数，符号位为1表示负数，其余各位表示数值部分。

如： $10000010, 00000010$

### 2、反码：

反码的定义如下：

(1)对于正数，它的反码表示与原码相同。即 $[x]_{\text{反}} = [x]_{\text{原}}$

(2)对于负数，则除符号位仍为“1”外，其余各位“1”换成“0”，“0”换成“1”，即得到反码 $[X]_{\text{反}}$ 。例如 $[-1101001]_{\text{反}} = 10010110$ 。

(3)对于0，它的反码有两种表示： $[+0]_{\text{反}} = 00\cdots0$   $[-0]_{\text{反}} = 11\cdots1$

### 3、补码：

正数的补码就是该正数本身。

$[01100100]_{\text{补}} = 01100100$

对于负数：两头的1不变，中间取反。（负数取反加一）

$[10100100]_{\text{补}} = 11011100$

$[+0]_{\text{补}} = [-0]_{\text{补}} = 00\cdots0$ 。

### 4、BCD 码（8421 码）

BCD 码就是用二进制代码表示的十进制数，也称 BCD 数。它是用 4 位二进制代码 0000—1001 来表示十进制数 0---9。如：39 的 BCD 码为 00111001。

## 三、整数和浮点数

### 1、整数

整型值可以用十进制，十六进制或八进制符号指定，前面可以加上可选的符号（- 或者 +）。

### 2、浮点数

浮点数，在计算机中用以近似表示任意某个实数。具体来说，这个实数由一个整数

或定点数（即尾数）乘以某个基数（计算机中通常是 2）的整数次幂得到，这种表示方法类似于基数为 10 的科学记数法。用 E(e) 来表示指数部分。如 123.456 或 123e-2。

#### 四、ASCII 码

American Standard Code for Information Interchange)

美国标准信息交换代码

将每个字符用 7 位的二进制数来表示，共有 128 种状态

大小字母、0...9、其它符号、控制符

‘ 0 ’    — —    48

‘ A ’    — —    65

‘ a ’    — —    97

#### 五、信息存储单位

(1)位 (bit, 缩写为 b): 度量数据的最小单位，表示一位二进制信息。

(2)字节(byte, 缩写为 B): 一个字节由八位二进制数字组成 (1 byte=8bit)。字节是信息存储中最常用的基本单位。

计算机存储器（包括内存与外存）通常也是以多少字节来表示它的容量。常用的单位有：

KB    1K=1024            MB    1M=1024K

GB    1G=1024M        TB    1T=1024G

3、机器字 (word): 字是位的组合，并作为一个独立的信息单位处理。字又称为计算机字，它取决于机器的类型、字长以及使用者的要求。常用的固定字长有 8 位、16 位、32 位等。

## 计算机网络的基本概念

### 一、计算机网络分类

计算机网络的分类方式有很多种,可以按地理范围、拓扑结构、传输速率和传输介质等分类。

地理范围分类:

- ① 局域网LAN, ②城域网MAN, ③广域网WAN, 广域网地理范围一般在几千公里左右,属于大范围连网。如几个城市,一个或几个国家,是网络系统中的最大型的网络,能实现大范围的资源共享,如国际性的Internet 网络。

按传输介质分类

传输介质是指数据传输系统中发送装置和接受装置间的物理媒体,按其物理形态可以划分为有线和无线两大类。

①有线网, 光纤和双绞线, 10M/100M/1000M

②无线网, 802. a b c n 协议。

### 二、计算机网络体系结构的核心是OSI模型

国际标准化组织 (ISO) 提出的开放系统互联参考模型 (OSI) 已成为网络体系结构的标准

(1)网络协议

常见的网络协议有IPX/SPX, TCP/IP等。

(2)网络互联模型

国际标准化组织ISO( International Standardization Organization)于1981年推出“开放系统互联结构模型”即OSI(OpenSystem Interconnection)标准。OSI不是一个实际的物理模型,而是一个将网络协议规范化了的逻辑参考模型。图1.6.3是OSI 七层模型图。

应用层
表示层
会话层
传输层
网络层
数据链路层
物理层

图 1.6.3 OSI 七层参考模型

通常把计算机网络分成通信子网和资源子网两大部分。OSI 参考模型的低三层：物理层、数据链路层和网络层归于通信子网的范畴；高三层：会话层、表示层和应用层归于资源子网的范畴。传输层起着承上启下的作用。

### 三、Internet 网络地址 (IP 地址)

现在的Internet最早起源于60 年代末期美国国防部的ARPAnet（阿帕网）。

通常一个IPv4地址共有32 位，分为4 段，每段8 位（也即1 个字节）。它的表示方法如下：xxx, xxx, xxx, xxx，其中每段的取值范围为0~255。IP地址是Internet上主机的一种数字标识，它由两部分组成，一部分是网络标识(netid)，另一部分是主机标识(hostid)。

第一段取值在1~127之间，表示主机所在的网络属于大型网（A类网），其\_\_\_\_\_值就是网络的网络号，后三段数字表示该主机号；

第一段数字取值在128~191 之间，表示主机所在网络为中型网（B 类网），第一段和第二段的数字联合表示该网络的网络号，第三段数字则表示子网号，第四段则是该主机号；

第一段数字取值为192~223 的，表示该主机所在的网络为小型网（C 类网），第一、二、三段数字的组合表示该网络的网络号，第四段是主机号。网站的IP地址就是203.207.226.84，则表示它的主机是属于C 类网，203.207.226是它所在网络的网络号，其主机号为84。

什么是IPv6、保留地址和域名？

## 逻辑运算

◆ 运算：与或非，异或

◆ 运算的优先级：非>与>或

### 1、“与” 运算（“•”，“∧”，and）

在逻辑问题中，如果决定某一事件发生的多个条件必须同时具备，事件才能发生，则这种因果关系称之为“与”逻辑(并且)。“与”运算又称为逻辑乘，其运算符号为“•”，有时也用“∧”表示。

两变量“与”运算关系可表示为  $F = A \cdot B$  或者  $F = A \wedge B$

“与”运算的运算法则为：

$$0 \cdot 0 = 0 \quad 1 \cdot 0 = 0$$

$$0 \cdot 1 = 0 \quad 1 \cdot 1 = 1$$

结论：若A、B均为1，则F为1；否则，F为0

推广： $A \cdot 0 = 0$        $A \cdot 1 = A$

### 2、“或”运算（“+”，“∨”，or）

在逻辑问题的描述中，如果决定某一事件是否发生的多个条件中，只要有一个或一个以上条件成立，事件便可发生，则这种因果关系称之为“或”逻辑。“或”运算又称逻辑加，

其运算符号为“+”，有时也用“ $\vee$ ”表示。两变量“或”运算的关系可表示为  $F = A + B$  或者  $F = A \vee B$

“或”运算的运算法则为：

$$0 + 0 = 0 \qquad 1 + 0 = 1$$

$$0 + 1 = 1 \qquad 1 + 1 = 1$$

结论：仅当A、B均为0时，F才为0

推广： $A+0=A$   $A+1=1$

### 3. “非”运算 $\neg$ - NOT

在逻辑问题中，如果某一事件的发生取决于条件的否定，即事件与事件发生的条件之间构成矛盾，则这种因果关系称为“非”逻辑。“非”运算也叫求反运算或者逻辑否定。其运算符号为“-”，有时也用“ $\neg$ ”表示。

“非”运算的逻辑关系可表示为  $F = A$  或者  $F = \neg A$

“非”运算的运算法则为

$$0 = 1 \qquad 1 = 0$$

## 字符串和表达式

### 一、在C++语言中逻辑运算

与	&&	and
或		or
非	!	not

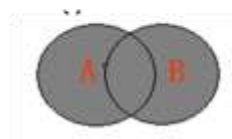
## 集合的运算

### 1、集合的运算：并、交、补、差

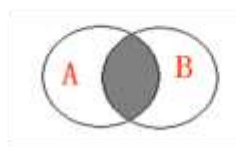
并： $\cup$

交： $\cap$  -

补： $\bar{\phantom{x}}$ 或 $\sim$ 或



$A \cup B$



$A \cap B$



$\bar{A}$



$A - B$

# 算法

## 一、递归

递归在计算机中的实现计算机执行递归算法时，是通过栈来实现的。

# 数据结构入门

## 一、复杂度：

算法分析，就是复杂度的问题。复杂度只算“最要命的”，比如，执行 $n^2$ 的算法前来个快排根本不拖速度， $n^2$ 多的都豁出去了不在乎区区一个 $n\log n$ 。书里对复杂度进行了严格的定义，包括 $O()$ 、 $o()$ 、 $\Theta()$ 、 $\Omega()$ 四种符号。简单地说， $O(n^2)$ 就是顶破天了搞个 $n^2$ 次； $o(n^2)$ 就是天花板不到 $n^2$ ，比 $n^2$ 矮一点（比如希尔排序就是 $o(n^2)$ ，因为它再倒霉也达不到 $n^2$ ）； $\Omega(n^2)$ 就是说某个算法随便怎么至少都要耗费 $n^2$ ，比如所有基于比较的排序都是 $\Omega(n\log n)$ ； $\Theta(n^2)$ 就是说它即是 $O(n^2)$ 又是 $\Omega(n^2)$ ，被天花板和水泥地夹在中间了，动不了了，就是它了。这里面有一个经典的例子，就是最大子序列（找数列中连续一段数之和最大）的四种算法，复杂度分别为 $O(n^3)$ 、 $O(n^2)$ 、 $O(n\log n)$ 和 $O(n)$ 。

附希腊字母读音： $O$ “大欧”、 $o$ “小欧”、 $\Theta$ “西塔”、 $\Omega$ “欧米伽”。

## 二、表、栈和队列：

表、栈和队列是三个基本的数据结构。说穿了表就是把数据找起来排排坐吃果果，找什么东西都来把整个队伍找一遍。

栈就是一个桶，后放进去的先拿出来，它下面本来有的东西要等它出来之后才能出来，就好像你看到了一个丑人不可能今天的中饭还没吐出来就先把早饭吐出来了。栈是拿来模拟多个过程的调用的（比如递归），实际点的用途就是表达式计算。

队列好比堵车，先进去的先出来。先进队先买票，不能插队。常拿来实现广搜。（BaihowFF注：广搜就是一般说的广度优先搜索，即BFS）

# 二叉树

## 一、二叉树定义：

是另一种树形结构，他的特点是每个节点至多只有两棵子树，并且，二叉树的子树有左右之分，其次序不能任意颠倒。

## 二、二叉树的性质：

性质1：在二叉树的第 $i$ 层上至多有 $2^{i-1}$ 个结点。（ $i \geq 1$ ）

性质2：深度为 $k$ 的二叉树上至多含 $2^k - 1$ 个结点（ $k \geq 1$ ）

性质3：对任何一棵二叉树，若它含有 $n_0$ 个叶子结点、 $n_2$ 个度为2的结点，则必存在关系式： $n_0 = n_2 + 1$ 。通俗点说：即叶子节点比二叉节点（分支节点）数多一个。

**满二叉树：**叶结点个数为 $N$ ，则它的结点总数为 $2 * N - 1$ 。

**完全二叉树：**在满二叉树的最底层自右向左依次去掉若干个节点得到的二叉树。

性质4： $n$ 层完全二叉树，深度为 $N$ ，节点数为 $(2^N) - 1$ ，叶子节点为 $2^{(N-1)}$ ， $2^N$ 表示2的 $N$ 次方。



### 三、二叉树的遍历：

三种遍历的顺序分别是这样的：

- ① 前序遍历：根-左-右
- ② 中序遍历：左-根-右
- ③ 后序遍历：左-右-根

### 四、经典问题：

若某二叉树的前遍历访问顺序是序abdgcefh, 中序遍历顺序是dgbaeCHF, 则后序遍历的访问顺序是什么。

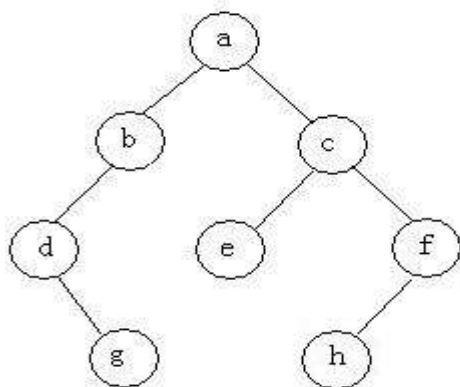
解答：此题的解答过程如下：

(1) 由前序遍历结果我们可知a为根结点，再看中序遍历结果，因为中序遍历顺序是左子树、根、右子树，因此由“中序遍历顺序是dgbaeCHF”可断定，dgb为该二叉树的左子树中序遍历结果，eCHF为右子树中序遍历结果。

(2) 由前序遍历结果可知，左子树的前序遍历结果是bdg, 右子树的前序遍历结果是cefh; 因此，和第一步分析类似，可知b为左子树的根，再由“dgb为该二叉树的左子树中序遍历结果”可知，dg为该左子树的左子树的中序遍历结果，再由dg在前序遍历结果中排列顺序dg可知，d为根，因此由“dg为该左子树的左子树的中序遍历结果”可推出g为d的右孩子。

到此为止，可以完全推断出该二叉树的左子树的结构了。

按照同样方法，可以推断出该二叉树的右子树的结构，因此整个二叉树的结构图如下：



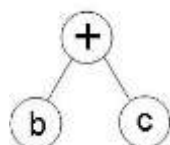
据此图，不难看出该二叉树的后序遍历结果是：gdbehfca.

### 五、按层次遍历二叉树

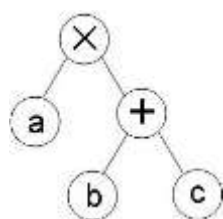
自然表达式转换为前/中/后缀表达式，其实是很简单的。首先将自然表达式按照优先级顺序，构造出与表达式相对应的二叉树，然后对二叉树进行前/中/后缀遍历，即得到前/中/后缀表达式。

举例说明将自然表达式转换成二叉树： $a \times (b+c) - d$

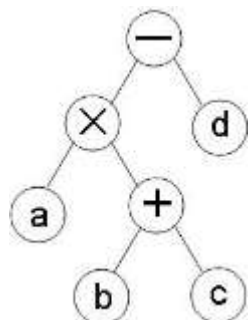
- ① 根据表达式的优先级顺序，首先计算  $(b+c)$ ，形成二叉树



- ② 然后是  $a \times (b+c)$ ，在写时注意左右的位置关系



③最后在右边加上-d



所以还是以刚才的这个例子，在最终二叉树的基础上可以得出：

前缀表达式：- \* a + b c d

中缀表达式：a \* b + c - d

后缀表达式：a b c + \* d -

## 排序与查找

### 一、各排序算法时间复杂度比较

算法名称	时间复杂度	一句话描述
冒泡排序	$O(n^2)$	交换相邻位置的元素，每次把最大（或最小）元素换到最右边，每次范围减一。
选择排序	$O(n^2)$	在未排序序列中找到最小元素，存放（交换任意两个元素）到排序序列的起始位置，再从剩余未排序元素中继续寻找最小元素，放到已排序序列末尾。要经过 $n-1$ 趟。（不稳定）
插入排序	$O(n^2)$	把第 $N$ 个元素插入前 $N-1$ 有序的数组中，从后往前扫描，不符合条件的元素往后移，符合条件就将第 $N$ 个元素放入当前位置。
快速排序	$O(n \log 2n)$ ， 最坏 $O(n^2)$	找到一个参考元素，将比它小的元素放到其左边，比它大的元素放到其右边，参考元素就在最终的位置，再对左右进行排序。（不稳定，优化方案：略）。
堆排序	$O(n \log n)$	略
归并排序	$O(n \log 2n)$	略

<http://www.sorting-algorithms.com/> <http://www.onlyblank.com/barcode/sort.html>

### 二、非比较排序算法

计数排序，基数排序，桶排序等非比较排序算法，平均时间复杂度都是 $O(n)$ 。这些排序因为其待排序元素本身就含有了定位特征，因而不需要比较就可以确定其前后位置，从而可以突破比较排序算法时间复杂度 $O(n \log n)$ 的理论下限。

### 三、查找

#### (1) 二分查找算法

二分查找算法先比较位于集合中间位置的元素与键的大小，有三种情况（假设集合是从小到大排列的）：

键小于中间位置的元素，则匹配元素必在左边（如果有的话），于是对左边的区域应用二分搜索。

键等于中间位置的元素，所以元素找到。

键大于中间位置的元素，则匹配元素必在右边（如果有的话），于是对右边的区域应用二分搜索。

注：比较次数 $n$ 次，可查找元素个数 $2^n - 1$ 个。

排序方法	时间复杂度（平均）	时间复杂度（最坏）	时间复杂度（最好）	空间复杂度	稳定性
插入排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
希尔排序	$O(n^{1.3})$	$O(n^2)$	$O(n)$	$O(1)$	不稳定
选择排序	$O(n^2)$	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
堆排序	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(1)$	不稳定
冒泡排序	$O(n^2)$	$O(n^2)$	$O(n)$	$O(1)$	稳定
快速排序	$O(n \log_2 n)$	$O(n^2)$	$O(n \log_2 n)$	$O(n \log_2 n)$	不稳定
归并排序	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n)$	稳定
计数排序	$O(n+k)$	$O(n+k)$	$O(n+k)$	$O(n+k)$	稳定
桶排序	$O(n+k)$	$O(n^2)$	$O(n)$	$O(n+k)$	稳定
基数排序	$O(n \cdot k)$	$O(n \cdot k)$	$O(n \cdot k)$	$O(n+k)$	稳定

## 信息技术的新发展

虚拟化技术：服务器虚拟化，存储与客户端虚拟化。

云计算：“云计算”提供数据存储和平台测试等网络服务。

客户端变革：虚拟化技术的流行，将为客户端计算机技术带来新的变革。

移动应用程序竞争：目前，苹果iPhone应用程序，谷歌Android明年将推出大量新机型，移动应用程序市场的竞争将进一步加剧。

实时搜索（Real-Time Search）：Amit Singhal正在Google领导着一个团队挖掘社会化网络的信息，提供与传统Web搜索相同质量和价值的信息。

## 附录 一、初赛内容与要求

计算机的基本知识	<ol style="list-style-type: none"> <li>1. 计算机和信息社会（信息社会的主要特征、计算机的主要特征、数字通信网络的主要特征、数字化）；</li> <li>2. 信息输入输出基本原理（信息交换环境、文字图形多媒体信息的输入输出方式）；</li> <li>3. 信息的表示与处理（信息编码、微处理部件 MPU、内存储结构、指令，程序，和存储程序原理、程序的三种基本控制结构）；</li> <li>4. 信息的存储、组织与管理（存储介质、存储器结构、文件管理、数据库管理）；</li> <li>5. 信息系统组成及互连网的基本知识（计算机组成原理、槽和端口的部件间可扩展互连方式、层次式的互连结构、互联网络、TCP/IP 协议、HTTP 协议、WEB 应用的主要方式和特点）；</li> <li>6. 人机交互界面的基本概念（窗口系统、人和计算机交流信息的途径（文本及交互操作））；</li> <li>7. 信息技术的新发展、新特点、新应用等。</li> </ol>	
计算机的基本操作	<ol style="list-style-type: none"> <li>1. Windows 和 LINUX 的基本操作知识；</li> <li>2. 互联网的基本使用常识（网上浏览、搜索和查询等）；</li> <li>3. 常用的工具软件使用（文字编辑、电子邮件收发等）。</li> </ol>	
程序设计基本知识	数据结构的类型	<ol style="list-style-type: none"> <li>1. 程序语言中基本数据类型（字符、整数、长整数、浮点）；</li> <li>2. 浮点运算中的精度和数值比较；</li> <li>3. 一维数组（串）与线性表；</li> <li>4. 记录类型（PASCAL）/ 结构类型（C）。</li> </ol>
	程序设计	<ol style="list-style-type: none"> <li>1. 结构化程序设计的基本概念；</li> <li>2. 阅读理解程序的基本能力；</li> <li>3. 具有将简单问题抽象成适合计算机解决的模型的基本能力；</li> <li>4. 具有针对模型设计简单算法的基本能力；</li> <li>5. 程序流程描述（自然语言/伪码/NS 图/其他）；</li> <li>6. 程序设计语言（PASCAL/C/C++）。</li> </ol>
	基本算法处理	<ol style="list-style-type: none"> <li>1. 初等算法（计数、统计、数学运算等）；</li> <li>2. 排序算法（冒泡法、插入排序、合并排序、快速排序）；</li> <li>3. 查找（顺序查找、二分法）；</li> <li>4. 回溯算法。</li> </ol>