

软件设计师考试常用公式

一． 计算机组成与结构

1. 数制的转换。

(1) 二进制与八进制

$$(1100100)_2 = (001\ 100\ 100)_2 = (1\ 4\ 4)_8$$

(2) 二进制与十进制

$$(1100100)_2 = 1*2^6 + 1*2^5 + 1*2^2 = (100)_{10}$$

(3) 二进制与十六进制

$$(1100100)_2 = (0110\ 0100)_2 = (64)_{16}$$

2. 逻辑代数的基本运算。

基本的逻辑运算有“与”、“或”、“非”、“异或”。

常用的逻辑运算公式：

$$\text{交换律：} A+B=B+A \quad A*B=B*A$$

$$\text{结合律：} A+(B+C) = (A+B)+C$$

$$\text{分配律：} A*(B+C) = A*B+A*C \quad A+(B*C) = (A+B)*(A+C)$$

$$\text{反演律：} \overline{A+B} = \overline{A} * \overline{B}$$

$$\text{重叠律：} A+A=A \quad A*A=A$$

$$\text{互补律：} A * \overline{A} = 0 \quad A + A = 1$$

$$\text{对合律：} \overline{\overline{A}} = A$$

$$\text{0-1律：} 0+A=A \quad A*A=0$$

3. 计算机的系统可靠性。

计算机系统的可靠性是指从它开始运行 ($t=0$) 到某个时刻 t 这段时间内能正常运行的概率, 用 $R(t)$ 表示。
失效率则是指单位时间内失效的元件数与元件总数的比例, 以 λ 表示。当为常数时, 可靠性与失效率的关系为：

$$R(t) = e^{-\lambda t}$$

两次故障之间系统能正常工作的时间的平均值称为平均无故障时间： $MTBF=1/\lambda$

通常用平均修复时间 (MTRF) 来表示计算机的可维修性, 即计算机的维修效率, 指从故障发生到机器修复平均所需要的时间。计算机的可用性是指计算机的使用效率, 它以系统到执行任务的任意时刻能正常工作的概率 A 来表示：

$$A = \text{MTBF} / (\text{MTBF} + \text{MTTR})$$

计算机的RAS技术，就是指用可靠性R、可用性A和可维修性S三个指标来衡量一个计算机系统。

常见的计算机系统可靠性数学模型有3种。

串联系统：

该系统由N个子系统组成，当且仅当所有的子系统都能正常工作时，系统才能正常工作。整个系统的可靠性R和失效率μ分别为：

$$\text{可靠性: } R = R_1 R_2 \dots R_N$$

$$\text{失效率: } \mu = \mu_1 + \mu_2 + \dots + \mu_N$$

并联系统：

该系统由N个子系统组成，只要有一个子系统正常工作，系统就能正常运行。整个系统的可靠性R和失效率μ分别为：

$$\text{可靠性: } R = 1 - (1 - R_1)(1 - R_2) \dots (1 - R_N)$$

$$\mu = \frac{1}{\frac{1}{\mu_1} + \frac{1}{\mu_2} + \dots + \frac{1}{\mu_N}}$$

失效率：

N模冗余系统：

该系统由N个(N=2n+1)相同的子系统和一个表决器组成，表决器把N个子系统中占多数相同结果的输出作为系统的输出。N模冗余系统的可靠性为：

$$\text{可靠性: } R = \sum_{i=n+1}^N \binom{N}{i} R_0^i (1 - R_0)^{N-i} \quad \left(\text{其中 } \binom{N}{j} \text{ 表示从 } N \text{ 个元素中取 } j \text{ 个元素的组合数} \right)$$

计算机可用性：是指该系统在某一时刻提供有效使用的程度，以可用度A表示，可用度是在任意指定时刻系统能正确运行的概率。一般情况下，系统发生故障是可以修复的，可用下列公式计算： $A = \text{MTBF} / (\text{MTBF} + \text{MTTR})$ ，其中MTBF为平均无故障时间，MTTR是平均修复时间。平均修复时间是指多次故障中，从开始失效到系统修复所用的平均时间，可用下式计算：

$$\text{系统修复时间} = \text{申请维修时间} + \text{等待时间} + \text{维修时间} + \text{恢复时间}$$

4. 磁盘的容量

辅助存储器方面的计算：

1. 存储容量为 $\text{capacity} = n * t * s * b$, n为存放数据的总盘面数；t为每面的磁道数；s为每道的扇区数；b为每个扇区存储的字节数(通常为512字节)
2. 非格式化容量 = 磁道长度 * 位密度 * 总的磁道数
3. 寻道时间为磁头移动到目标磁道所需的时间。
4. 等待时间为待读写的扇区旋转到磁头下方所用的时间。一般用磁道旋转一周所用的时间的一半作为平均等待时间。
5. 磁盘存取时间 = 寻道时间 + 等待时间。
6. 位密度：沿磁道方向，单位长度存储二进制信息的个数；
7. 道密度：沿磁盘半径方向，单位长度内磁道的数目；
8. 数据传输速率 $R = B/T$, B为一个磁道上记录的字节数，T为每转一周的时间
9. 磁带机的容量计算：(这些公式要熟悉记住)
数据传输率 = 磁带记录密度 * 带速；

数据块长度 = 字节数 * 块因子 / 记录密度 + 块间间隔;
 读 N 条记录所需时间 = 启停时间 + 有效时间 + 间隔时间;

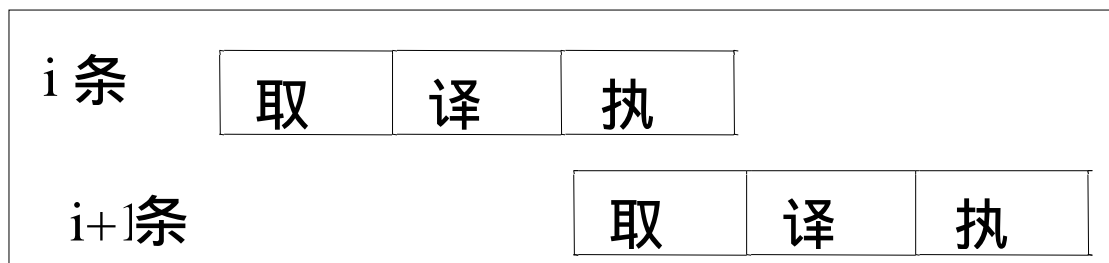
5. 流水线技术。

(1) 顺序执行

执行 K 条指令的时间 = $k * (t_{\text{取}} + t_{\text{译}} + t_{\text{执}})$

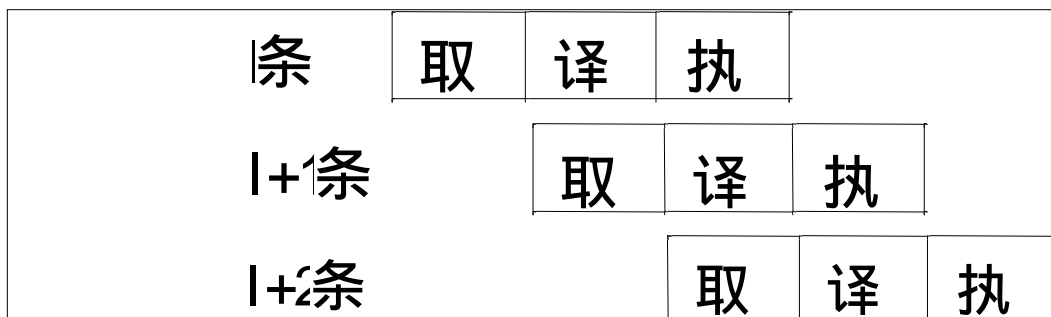
(2) 两条重叠

执行 k 条指令的时间 = $t_{\text{取}} + k * t_{\text{译}} + (k-1) * (t_{\text{取}}, t_{\text{执}})_{\max} + t_{\text{执}}$



(3) 三条重叠

执行 k 条指令的时间 = $t_{\text{取}} + (t_{\text{译}}, t_{\text{取}})_{\max} + (k-2) * (t_{\text{取}}, t_{\text{译}}, t_{\text{执}})_{\max} + (t_{\text{执}}, t_{\text{译}})_{\max} + t_{\text{执}}$



6. 校验码

(1) 海明校验 (下载资料)

(2) CRC 循环校验 (下载资料)

二. 编译原理

1. 正规式和它所表示的正规集

(1) 正规式定义中的“|”读为“或”(也有使用“+”代替“|”的);“.”读为“连接”;“*”读为“闭包”(即,任意有限次的自重复连接)。在不致混淆时,括号可省去,但规定算符的优先顺序为“(、”)”、“*”、“.”、“|”。连接符“.”一般可省略不写。“*”、“.”和“|”都是左结合的。

令 $S = \{a, b\}$, 上的正规式和相应的正规集的例子有:

正规式	正规集
a	{a}

$a b$	$\{a,b\}$
ab	$\{ab\}$
$(a b)(a b)$	$\{aa,ab,ba,bb\}$
a^*	$\{ \quad, a, a, \dots \dots \text{任意个 } a \text{ 的串} \}$
$(a b)^*$	$\{ \quad, a, b, aa, ab \dots \dots \text{所有由 } a \text{ 和 } b \text{ 组成的串} \}$
$(a b)^*(aa bb)(a b)^*$	$\{ \quad \text{*上所有含有两个相继的 } a \text{ 或两个相继的 } b \text{ 组成的串} \}$

(2) 设 r,s,t 为正规式，正规式服从的代数规律有：

$r s=s r$	"或"服从交换律
$r (s t)=(r s) t$	"或"的可结合律
$(rs)t=r(st)$	"连接"的可结合律
$r(s t)=rs rt$	$(s t)r=sr tr$ 分配律
$r=r, r \quad =r$	是"连接"的恒等元素零一律
$r r=r \quad r^*= \quad r rr \dots$	"或"的抽取律

2、几种文法的区别

文法类型	文法名称	语言名称	对应的自动机
0型 (PSG)	短语结构文法	递归可枚举语言	图灵机 (Turing)
1型 (CSG)	上下文有关文法	上下文有关语言	线性界限自动机
2型 (CFG)	上下文无关文法	上下文无关语言	非确定下推自动机
3型	右线性文法 (正规文法)	有限状态文法	有限状态自动机

三．操作系统

1．信号量和 P-V 操作 (参考下载资料)

P 操作

- I. $S-1 \quad S$
- II. 如果 $S < 0$ ，则该进程进入等待状态；否则继续进行

V 操作

- I. $S+1 \quad S$
 - II. 如果 $S = 0$ ，则唤醒队列中的一个等待进程
- 进程互斥的情况初值是 1，而同步的初值是 0

2．页式存储管理

绝对地址的计算公式如下： $\text{绝对地址} = \text{块号} * \text{块长} + \text{页内地址}$

位示图：

位示图中的每一位与一个主存块对应，每一位的值可以是“0”或“1”，当取值为“0”时表示对应块为空闲，当取值为“1”时表示对应块已被占用。在位示图中增加一个字节记录当前剩余的空闲块数当需要分配一块主存空间时，可从位示图中找一个为“0”的位，把该位置成占用标志“1”，且根据它在位示图中的字号和位号，按如下公式计算出与它对应的块号：

$$\text{块号} = \text{字号} \times \text{字长} + \text{位号}$$

这个块号就是当前所分配用来装作业信息的主存块的块号。

当一个作业执行结束时，应回收该作业所占用的主存块。根据归还的块号可计算出归还块在位示图中

的对应位置，计算公式如下，并将该位的占用标志置为“0”。

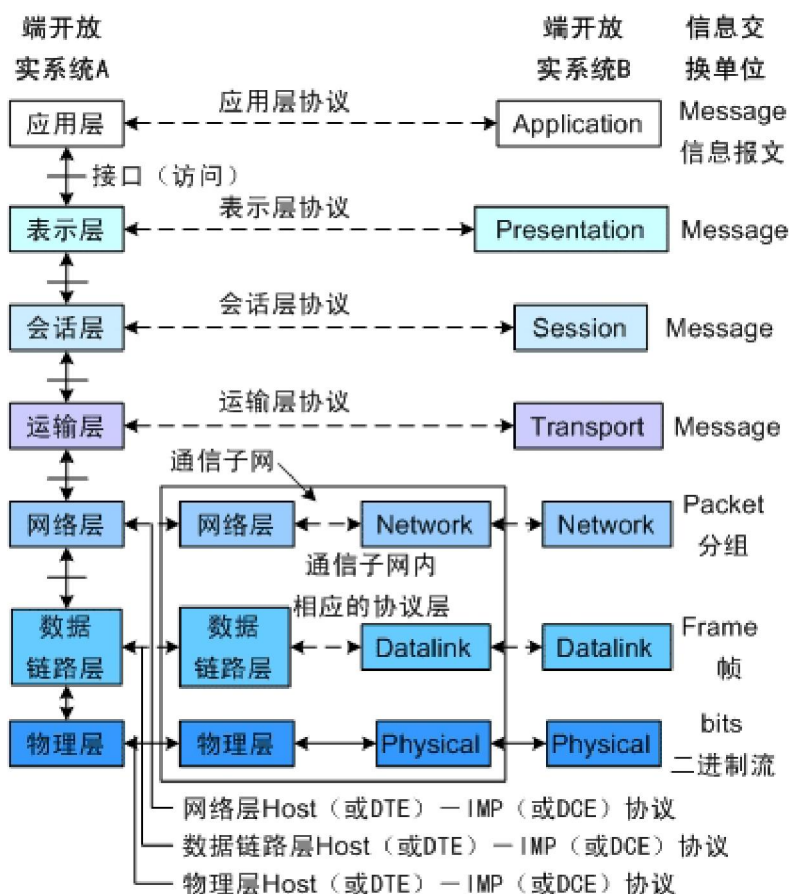
字号 = [块号 / 字长] , 位号 = 块号 mod 字长

四、多媒体技术

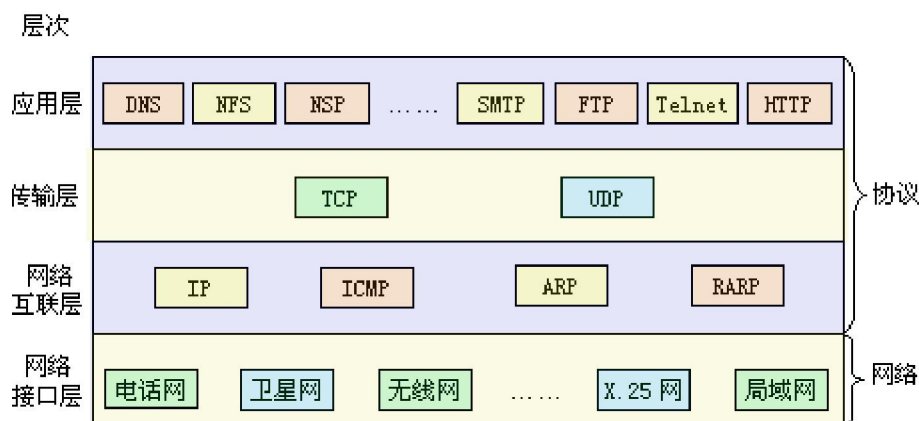
1. 音频采集计算：声音文件的存储量 = 采样频率 × 采样位数 × 声道数
2. 视频图象的容量计算：图像文件的存储量 = 分辨率 × 色彩数 (位)

五、计算机网络

1. ISO/OSI 基本参考模型



2. TCP/IP模型中的协议与网络



3. IP 地址与子网掩码

A 类网络 (1~126) 缺省子网掩码: 255 0 0 0

255 0 0 0 换算成二进制为 11111111 00000000 00000000 00000000

可以清楚地看出前 8 位是网络地址, 后 24 位是主机地址, 也就是说, 如果用的是标准子网掩码, 看第一段地址即可看出是不是同一网络的。如 21.0.0.1 和 21.240.230.1, 第一段为 21 属于 A 类, 如果用的是默认的子网掩码, 那这两个地址就是一个网段的。

B 类网络 (128~191) 缺省子网掩码: 255 255 0 0

C 类网络 (192~223) 缺省子网掩码: 255 255 255 0

主机号全“0”全“1”的地址在 TCP/IP 协议中有特殊含义 (广播地址和本网地址)

六、数据库

1. 并发控制的封锁:

一级封锁协议是: 事务 T 在修改数据 R 之前必须先对其加 X 锁, 直到事务结束才释放。事务结束包括正常结束 (COMMIT) 和非正常结束 (ROLLBACK)。

二级封锁协议是: 一级封锁协议加上事务 T 在读取数据 R 之前必须先对其加 S 锁, 读完后即可释放 S 锁。二级封锁协议除防止了丢失修改, 还可进一步防止读“脏”数据。

三级封锁协议是: 一级封锁协议加上事务 T 在读取数据 R 之前必须先对其加 S 锁, 直到事务结束才释放。三级封锁协议除防止了丢失修改和不读“脏”数据外, 还进一步防止了不可重复读。

2. 将 E - R 模型转换为关系模型时, 需要遵循的转换规则如下:

(1). 每个实体转换为一个关系。

(2). 一个一对一的联系可转换为一个关系模式, 将两端关系的码及联系的属性均作为该关系的属性, 任意一端的码作为该关系的码; 也可将任意一端的码及联系的属性合并到另一端实体集所转换的关系模式中。

(3). 一个一对多联系可转换为一个关系模式, 将两端关系的码及联系的属性均作为该关系的属性, “多”端的码作为该关系的码; 也可将“一”端的码及联系的属性合并到“多”端实体集所转换的关系模式中。

(4). 一个多对多联系应转换为一个关系模式, 两端的码及联系的属性为关系的属性, 两端的码共同组合为该关系的码。

3 个或 3 个以上多对多的联系应转换为一个关系，各关系的码及联系的属性为关系的属性，各端的码共同组合为该关系的码。

3. Armstrong 公理，如何在给定函数依赖集 F 的情况下方便地计算它的闭包。

Armstrong 公理的基本定律包括 3 条：

自反律：若 $X \rightarrow Y$ ，则 $X \rightarrow Y$ ，则

增补律：若 $X \rightarrow Y$ ，则 $X \rightarrow YZ$ ，则

传递律：若 $X \rightarrow Y$ 及 $Y \rightarrow Z$ ，则 $X \rightarrow Z$ ，则

上述推理规则是保真的，也是完备的，也就是说，根据这三个定律推出的函数依赖肯定是正确的函数依赖，而所有的函数依赖都可以根据这三个定律推出。但为了计算或证明的方便，又扩充了以下三个定律：

合并律：若 $X \rightarrow Y$ 及 $X \rightarrow Z$ ，则 $X \rightarrow YZ$ ，则

分解律：若 $X \rightarrow YZ$ ，则 $X \rightarrow Y$ 及 $X \rightarrow Z$ ，则

伪传递律：若 $X \rightarrow Y$ 及 $YZ \rightarrow W$ ，则 $X \rightarrow W$ ，则

4. 范式

(1) 1NF (第一范式)

对一个关系模式来说，如果它的每一个属性都是不可再分割的（原子的），这样的关系模式就属于第一范式，简称 1NF。

(2) 2NF (第二范式)

若关系模式 R 属于 1NF，且每一个非主属性都完全函数依赖于主码，则 R 属于第二范式，简称 2NF。

(3) 3NF (第三范式)

对于关系模式 R 2NF 来说，如果 R 中不存在这样的主码 X、属性集 Y 及非主属性 Z(ZY)，使得 $X \rightarrow Y$ 、 $Y \rightarrow Z$ 成立，则 R 属于第三范式，简称 3NF。

5. 规范化

关系数据库中的关系是要满足一定要求的，满足不同程度要求的属于不同的范式。一般来说：

5NF 4NF BCNF 3NF 2NF 1NF。

七、数据结构

1. 线性表

(1) 线性表的第 i 个元素 $a[i]$ 的存储位置可以使用以下公式求得：

$$LOC(a_i) = LOC(a_1) + (i-1) * l$$

式中 $LOC(a_1)$ 是线性表的第一个数据元素 a_1 的存储位置，通常称做线性表的起始位置或基地址。

2. 循环队列

队头、队尾指针加 1 时从 $maxSize - 1$ 直接进到 0，可用语言的取模(余数)运算实现。

队头指针进 1: $front = (front + 1) \% maxSize$;

队尾指针进 1: $rear = (rear + 1) \% maxSize$;

队列初始化: $front = rear = 0$;

队空条件: $front == rear$;

队满条件： $(\text{rear} + 1) \% \text{maxSize} == \text{front}$

3. 对角矩阵

对角矩阵中，所有的非零元素集中在以主对角线为中心的带状区域中，即除了主对角线和主对角线相邻两侧的若干条对角线上的元素之外，其余元素皆为零。当 $i-j > 1$ 时，元素 $a[i,j]=0$ 。

$$\text{LOC}(i,j)=\text{LOC}(0,0)+[3*i-1+(j-i+1)]=\text{LOC}(0,0)+(2i+j)$$

4. 三角矩阵可压缩存储到向量 $\text{sa}[0..n(n+1)/2]$ 中， $\text{sa}[k]$ 和 a_{ij} 的对应关系是：

$$k = \begin{cases} i(2n-i+1)/2 + j-i & \text{当 } i \leq j \text{ 时} \\ n(n+1)/2 & \text{当 } i > j \text{ 时} \end{cases}$$

5. 二叉树的性质：

二叉树具有下列重要特性。

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

深度为 k 的二叉树至多有 $2^k - 1$ 个结点 ($k \geq 1$)。

对任何一棵二叉树 T ，如果其终端结点数为 n_0 ，度为 2 的结点数为 n_2 ，则 $n_0 = n_2 + 1$ 。

具有 n 个结点的完全二叉树的深度为 $\lfloor \log_2 n \rfloor + 1$ 。

6. 哈夫曼及其应用：又称为最优树，是一类带权路径长度最短的树。

路径长度：从树中一个节点到另一个节点之间的分支构成的这两个节点之间的路径，路径上的分支树木就称为路径长度；

树的路径长度：从树根到每一节点的路径长度之和；

树的带权路径长度：树中所有叶子节点的带权路径长度之和；

哈夫曼树就是一棵 n 个叶子节点的二叉树，所有叶子节点的带权之和最小。

7. 有 $\frac{1}{2}n(n-1)$ 条边的无向图称为无向完全图，而具有 $n(n-1)$ 条弧的有向图成为有向完全图。

8. 如何评价查找算法的时间效率？由于查找算法中为确定其关键字等于给定值的数据元素的基本操作为“关键字和给定值相比”，因此通常以查找过程中关键字和给定值比较的平均次数作为比较查找算法的度量依据。

定义：查找过程中先后和给定值进行比较的关键字个数的期望值称作查找算法的平均查找长度。

对于含有 n 个记录的查找表，查找成功时的平均查找长度为

$$ASL = \sum_{i=1}^n p_i C_i$$

其中： p_i 为查找表中第 i 个记录的概率，且

$$\sum_{i=1}^n p_i = 1$$

C_i 为找到表中第 i 个记录（其关键字等于给定值）时，曾和给定值进行过比较的关键字的个数，显然， C_i 的值将随查找过程的不同而不同。

例如：

在顺序表中进行(顺序)查找查找成功的平均查找长度为：

$$ASL_{ss} = \sum_{i=1}^n p_i \times C_i = \sum_{i=1}^n p_i \times (n-i+1) \\ = p_1 \times n + p_2 \times (n-1) + \cdots + p_n \times 1$$

若查找表中每个记录的查找概率相等，即

$$p_i = \frac{1}{n}$$

则等概率查找时顺序查找的平均查找长度为

$$ASL_{ss} = \frac{1}{n} \sum_{i=1}^n (n-i+1) = \frac{n+1}{2}$$

9. 排序总表

排序方法		简介	平均时间	最坏情况	辅助存储	是否稳定
简单排序	选择排序	反复从还未排好序的那部分线性表中选出键值最小的结点，并按从线性表中选出的顺序排列结点，重新组成线性表。直至未排序的那部分为空，则重新形成的线性表是一个有序的线性表。(单选择排序)	$O(n^2)$	$O(n^2)$	$O(1)$	不稳定
	直接插入排序	假设线性表的前面 i 个结点序列 e_0, e_1, \dots, e_{i-1} 是已排序的。对结点在这有序结点 e_i 序列中找插入位置，并将 e_i 插入，而使 $i+1$ 个结点序列 e_0, e_1, \dots, e_i 也变成排序的。依次对 $i=1, 2, \dots, n-1$ 分别执行这样的插入步骤，最终实现线性表的排序。	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
	冒泡排序	对当前还未排好序的范围内的全部结点，自上而下对相邻的两个结点依次进行比较和调整，让键值大的结点往下沉，键值小的结点往上冒。即，每当两相邻比较后发现它们的排列顺序与排序要求相反时，就将它们互换。	$O(n^2)$	$O(n^2)$	$O(1)$	稳定
希尔排序		对直接插入排序一种改进，又称“缩小增量排序”。先将整个待排序列分割成为若干子序列分别进行直接插入排序，待整个序列中的记录“基本有序”时，再对全体记录进行一次直接插入排序。如果待排序记录序列为“正序”时，复杂度可达到 $O(n)$	$kn \ln n$	$O(n^2)$	$O(\log n)$	不稳定
快速排序		对冒泡排序的一种本质的改进。通过一趟扫描后，使得待排序序列的长度能大幅度的减少。在一趟扫描后，使某个结点移到中间的	$O(n \log n)$	$O(n^2)$	$O(\log n)$	不稳定

	正确位置，并使在它左边序列的结点的键值都比它的小，而它右边序列的结点的键值都不比它的小。称这样一次扫视为“划分”。每次划分使一个长序列变成两个新的较小子序列，对这两个小的子序列分别作同样的划分，直至新的子序列的长度为 1 使才不再划分。当所有子序列长度都为 1 时，序列已是排好序的了。				
堆排序（有两种情况：堆顶元素是最大值和堆顶元素是最小值）	一种树形选择排序，是对直接选择排序的有效改进。一个堆是这样一棵顺序存储的二叉树，它的所有父结点（ $e[i]$ ）的键值均不小于它的左子结点（ $e[2*i+1]$ ）和右子结点（ $e[2*i+2]$ ）的键值。初始时，若把待排序序列的 n 个结点看作是一棵顺序存储的二叉树，调整它们的存储顺序，使之成为一个堆，这时堆的根结点键值是最大者。然后将根结点与堆的最后一个结点交换，并对少了一个结点后的 $n-1$ 结点重新作调整，使之再次成为堆。这样，在根结点得到结点序列键值次最大值。依次类推，直到只有两个结点的堆，并对它们作交换，最后得到有序的 n 个结点序列。	$O(n\log n)$	$O(n\log n)$	$O(1)$	不稳定
归并排序	将两个或两个以上的有序子表合并成一个新的有序表。对于两个有序子表合并一个有序表的两路合并排序来说，初始时，把含 n 个结点的待排序序列看作有 n 个长度都为 1 的有序子表所组成，将它们依次两两合并得到长度为 2 的若干有序子表，再对它们作两两合并……直到得到长度为 n 的有序表，排序即告完成。	$O(n\log n)$	$O(n\log n)$	$O(n)$	稳定

八．算法分析与设计

1．算法具有以下 5 个属性：

- 有穷性：一个算法必须总是在执行有穷步之后结束，且每一步都在有穷时间内完成。
- 确定性：算法中每一条指令必须有确切的含义。不存在二义性。只有一个入口和一个出口
- 可行性：一个算法是可行的就是算法描述的操作是可以通过已经实现的基本运算执行有限次来实现的。
- 输入：一个算法有零个或多个输入，这些输入取自于某个特定对象的集合。
- 输出：一个算法有一个或多个输出，这些输出同输入有着某些特定关系的量。

2．迭代法 穷举搜索法 递推法 递归法 贪婪法 分治法 动态规划法 回溯法

（参考：《软件设计师考试考点分析与真题详解（软件设计技术篇）》）

因时间仓促，总结的不够完整，欢迎各位看客不吝提出宝贵的意见，发送邮件到 **Email: school@csai.cn**，谢谢！