1、浮点数的表示

（1）浮点数格式

阶码决定范围，阶码越长，范围越大；

尾数决定精度，尾数越长，精度越高。

（2）浮点数运算过程

对阶→尾数计算→格式化；

对阶：小数像大数看齐，尾数右移。

2、海明校验码

检验方式	校验码位数	校验码位置	检错	纠错	校验方式
海明校验	$2^r \geqslant m+r+1$	插入在信息位中间	可检错	可纠错	分组奇偶校验

3、CPU 组成

CPU 主要由运算器、控制器、寄存器组和内部总线等部件组成。

4、流水线技术

流水线建立时间：第 1 条指令执行时间。

流水线周期：指令分段后，最长段时间。

流水线执行时间（默认使用理论公式，无答案时考虑实践公式）。

理论公式：流水线建立时间+（指令条数-1）*流水线周期。

实践公式：指令段数*流水线周期+（指令条数-1）*流水线周期。

吞吐率=指令条数/流水线执行时间。

最大吞吐率=流水线周期的倒数。

5、Cache

在计算机的存储系统体系中，Cache 是（除寄存器以外）访问速度最快的层次。解决 CPU 与主存之间速度容量不匹配问题。

6、输入输出技术

程序控制（查询）方式：分为无条件传送和程序查询方式。方法简单，硬件开销小，但 I/O 能力不高，严重影响 CPU 的利用率。

程序中断方式：与程序控制方式相比，中断方式因为 CPU 无需等待而提高了传输请求的响应速度。

DMA 方式：DMA 方式是为了在主存与外设之间实现高速、批量数据交换而设置的。DMA 方式比程序控制方式与中断方式都高效。

7、线程

同一个进程当中的各个线程，可以共享该进程的各种资源，如内存地址空间、代码、数据、文件等，线程之间的通信与交流非常方便。

对于同一个进程当中的各个线程来说， 他们可以共享该进程的大部分资源。每个线程都有自己独立的 CPU 运行上下文和栈，这是不能共享的（程序计数器、寄存器和栈不能共享）。

8、PV 操作

P 操作：S=S-1(申请并锁定资源)；S<0(检查资源是否足够)。

V 操作：S=S+1(释放资源)；S<=0(检查是否有进程排队并通知排队进程)。

S 信号量：表示资源数，初值即为初始状态无操作时，资源的数量；信号量小于 0 的时候，还可以表示排队的进程数量。

9、前趋图与 PV 操作分析题技巧

针对箭线标注信号量，箭线的起点位置是 V 操作（即前趋活动完成后以 V 操作通知后继活动）；箭线的终点位置是 P 操作（即后继活动开始前以 P 操作检查前趋活动是否完成）。

10、死锁

死锁四大条件：互斥、保持和等待、不剥夺、环路等待。

假设 m 个进程各自需要 w 个 R 资源，系统中共有 n 个 R 资源，此时不可能形成死锁的条件是：m*(w-1)+1<=n。

11、页式存储的淘汰原则

页面淘汰时，主要依据原则（考试中默认按照此原则进行淘汰）：先淘汰最近未被访问的（访问位为 0），其次多个页面访问位为 0 时，则淘汰未被修改的（即修改位为 0，因为修改后的页面淘汰时代价更大）。

12、数据库三级模式两级映像

外模式-视图；模式-基本表；内模式-文件。

外模式-模式映射，保证数据逻辑独立性，即数据的逻辑结构发生变化后，用户程序也可以不修改。只需要修改外模式和概念模式之间的映像。

模式-内模式映射，保证数据物理独立性，即当数据的物理结构发生改变时，应用程序不用改变。只需要修改概念模式和内模式之间的映像。

13、E-R 图转关系模式转换原则

实体必须单独转换为 1 个关系模式。

联系根据类型不同：

（1）一对一联系的转换有 2 种方式。

独立的关系模式：并入两端主键及联系自身属性。（主键：任一端主键）

归并（任意一端）：并入另一端主键及联系自身属性。（主键：保持不变）

（2）一对多联系的转换有 2 种方式。

独立的关系模式：并入两端主键及联系自身属性。（主键：多端主键）

归并（多端）：并入另一端主键及联系自身属性。（主键：保持不变）

（3）多对多联系的转换只有 1 种方式

独立的关系模式：并入两端主键及联系自身属性。（主键：两端主键的组合键）

14、关系代数

笛卡尔积×：结果的属性列数是二者之和，结果的元组行数是二者乘积。

投影π：对垂直方向的属性列进行筛选。

选择σ：对水平方向的元组行进行筛选。

自然连接⋈：结果的属性列数是二者之和减去重复列数，结果元组是同名属性列取值相等的元组。

15、规范化程度判断即范式判定依据

1NF：属性值都是不可分的原子值。（基本二维表）

2NF：在 1NF 基础上，消除了非主属性对候选键的部分函数依赖。（候选键是单属性至少满足 2NF）

3NF：在 2NF 基础上，消除了非主属性对候选键的传递函数依赖。（没有非主属性至少满足 3NF）

BCNF：在 3NF 基础上，消除了主属性对候选键的部分函数依赖和传递函数依赖。

16、TCP 与 UDP 区别

TCP 与 UDP 均支持对具体指定端口号进行通信。

但连接管理、差错校验、重传等能力只有 TCP 具备。

17、常见协议功能和默认端口

协议名	默认端口	功能	特殊说明
HTTP	80	超文本传输协议，网页传输	不安全，结合 SSL 的 HTTPS 协议是安全的超文本传输协议，默认端口 443
Telnet	23	远程协议	不安全，SSH 是安全的远程协议
FTP	20 数据 21 控制	文件传输协议	不安全，结合 SSL 的 SFTP 是安全的文件传输协议。

POP3	110	邮件收取	附加多媒体数据时需采用 MIME（MIME 不安全，结合 SSL 的 MIME/S 是安全的多媒体邮件协议）。使用 WEB 方式收发电子邮件时必须设置账号密码登录。
SMTP	25	邮件发送	
DNS	53	域名解析协议，记录域名与 IP 的映射关系	本地客户端主机首查本机 hosts 文件 域名服务器首查本地缓存
DHCP	67	IP 地址自动分配	169.254.X.X 和 0.0.0.0 是无效地址
SNMP	161	简单网络管理协议	服务器仅发送消息给当前团体
ARP	地址解析协议， IP 地址转换为 MAC 地址		ARP Request 请求采用广播进行传送 ARP Response 响应采用单播进行传送
RARP	反向地址解析协议， MAC 地址转 IP 地址		无
ICMP	因特网控制协议		PING 命令来自该协议
IGMP	组播协议		无

18、加密算法

常见对称密钥加密算法（共享密钥加密技术）：DES、 3DES(三重 DES)、 RC-5、IDEA、AES 算法。

常见非对称密钥加密算法（公开密钥加密技术）： RSA、ECC。

常见的摘要算法：MD5(128 位)，SHA(160 位)。

19、加密技术应用

数字信封：用接收方公钥加密使用的对称密钥。

数字签名：用发送方私钥签名，保证发送方身份真实性，发送者不可抵赖。与信息摘要结合，可防篡改。

信息摘要：单向散列值函数，防篡改，保证消息完整性。

数字证书

数字证书的内容包括：CA 签名、用户信息（用户名称）、用户公钥等。

证书中的 CA 签名验证数字证书的可靠性、验证网站真伪。

用户公钥：客户端利用证书中的公钥加密，服务器利用自己的私钥解密。

20、常见的软件开发模型（1）

（1）瀑布模型

容易理解，管理成本低，每个阶段都有对应的成果产物，各个阶段有明显的界限划分和顺序要求，一旦发生错误，整个项目推倒重新开始。

适用于需求明确的项目，一般表述为需求明确、或二次开发，或者对于数据处理类型的项目。

（2）V 模型

强调测试贯穿项目始终，而不是集中在测试阶段。是一种测试的开发模型。

（3）喷泉模型

以用户需求为动力，以对象为驱动，适合于面向对象的开发方法。

特点是迭代、无间隙。

21、常见的软件开发模型（2）

（1）原型模型

典型的原型开发方法模型。适用于需求不明确的场景，可以帮助用户明确需求。

（2）增量模型

可以有多个可用版本的发布，核心功能往往最先完成，在此基础上，每轮迭代会有新的增量发布，核心功能可以得到充分测试。强调每一个增量均发布一个可操作的产品。

（3）螺旋模型

典型特点是引入风险分析。结合了瀑布模型和演化模型的优点，最主要的特点在于加入了风险分析。它是由制定计划、风险分析、实施工程、客户评估这一循环组成的，它最初从概念项目开始第一个螺旋。

22、开发方法

结构化开发方法：用户至上，严格区分工作阶段，每阶段有任务和结果，强调系统开发过程的整体性和全局性，系统开发过程工程化，文档资料标准化，自顶向下，逐步求精。

原型开发方法：适用于需求不明确的情况。

面向对象开发方法：更好的复用性，关键在于建立一个全面、合理、统一的模型，分析、设计、实现三个阶段界限不明确。

23、模块设计原则

保持模块的大小适中。

尽可能减少调用的深度。

多扇入，少扇出。

单入口，单出口。

模块的作用域应该在模块之内【作用域在控制域内】。

功能应该是可预测的。

24、内聚性

内聚类型	描　　述
功能内聚	完成一个单一功能，各个部分协同工作，缺一不可
顺序内聚	处理元素相关，而且必须顺序执行
通信内聚	所有处理元素集中在一个数据结构的区域上
过程内聚	处理元素相关，而且必须按特定的次序执行
瞬时内聚（时间内聚）	所包含的任务必须在同一时间间隔内执行
逻辑内聚	完成逻辑上相关的一组任务
偶然内聚（巧合内聚）	完成一组没有关系或松散关系的任务

25、耦合性

耦合类型	描　　述
非直接耦合	两个模块之间没有直接关系，它们之间的联系完全是通过主模块的控制和调用来实现的
数据耦合	一组模块借助参数表传递简单数据
标记耦合	一组模块通过参数表传递记录信息（数据结构）
控制耦合	模块之间传递的信息中包含用于控制模块内部逻辑的信息
外部耦合	一组模块都访问同一全局简单变量，而且不是通过参数表传递该全局变量的信息
公共耦合	多个模块都访问同一个公共数据环境
内容耦合	一个模块直接访问另一个模块的内部数据；一个模块不通过正常入口转到另一个模块的内部；两个模块有一部分程序代码重叠；一个模块有多个入口

26、维护

更正性维护：针对真实存在并已经发生的错误进行的维护行为。

预防性维护：针对真实存在但还未发生的错误进行的维护。

适应性维护：指使应用软件适应信息技术变化和管理需求变化而进行的修改。企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。

完善性维护：扩充功能和改善性能而进行的修改。对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。

27、质量属性及其子特性

功能性：适合性、准确性、互操作性、安全保密性。

可靠性：成熟性、容错性、易恢复性。

易用性：易理解性、易学性、易操作性。

效率：时间特性、资源利用性。

维护性：易分析性、稳定性、易测试性、易改变性。

可移植性：适应性、易安装性、一致性、易替换性。

28、风险管理

（1）风险的特性：具有不确定性，可能会造成损失。

（2）风险的类别

项目风险涉及到各种形式的预算、进度、人员、资源以及客户相关的问题，并且可能导致项目损失。

技术风险涉及到技术相关的可能会导致项目损失的问题。

商业风险与市场因素相关。

社会风险涉及到政策、法规等因素。

（3）风险曝光度(RiskExposure)=错误出现率(风险出现率) X 错误造成损失(风险损失)。

29、面向对象基本概念

面向对象：对象+分类+继承+通过消息的通信。

对象：属性（数据）+方法（操作）+对象 ID。

封装：隐藏对象的属性和实现细节，仅对外公开接口（信息隐藏技术）。

类（实体类/控制类/边界类）：对对象的抽象。

接口：一种特殊的类，他只有方法定义没有实现。

继承与泛化：复用机制。

消息和消息通信：对象之间进行通信的一种构造叫做消息。消息是异步通信的。

重置/覆盖：在子类中重新定义父类中已经定义的方法。

重载：一个类可以有多个同名而参数类型不同的方法。

动态绑定：根据接收对象的具体情况将请求的操作与实现的方法进行连接（运行时绑定）。

多态：不同对象收到同样的消息产生不同的结果（软设一般只涉及过载多态-同一个名字在不同的上下文中所代表的含义不同）。

30、符号表和错误管理

（1）符号表

符号表的作用是记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成，在编译过程中需要对符号表进行快速有效地查找、插入、修改和删除等操作。符号表的存在可以贯穿编译所有阶段。

（2）错误管理

静态错误：编译时所发现的程序错误，分为语法错误和静态语义错误。

语法错误包含：单词拼写错误、标点符号错误、表达式中缺少操作数、括号不匹配等有关语言结构上的错误。

静态语义分析：运算符与运算对象类型不合法等错误。

动态错误：发生程序运行时，也叫动态语义错误。包括死循环、变量取零时做除数、引用数组元素下标越界等错误。

31、类图关系

依赖关系：一个事物发生变化影响另一个事物。

泛化关系：特殊/一般关系。

关联关系：描述了一组链，链是对象之间的连接。

聚合关系：整体与部分生命周期不同。

组合关系：整体与部分生命周期相同。

实现关系：接口与类之间的关系。

32、用例关系

包含关系:其中这个提取出来的公共用例称为抽象用例,而把原始用例称为基本用例或基础用例系，当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。

扩展关系:如果一个用例明显地混合了两种或两种以上的不同场景,即根据情况可能发生多种分支，则可以将这个用例分为一个基本用例和一个或多个扩展用例，这样使描述可能更加清晰。

泛化关系:当多个用例共同拥有一种类似的结构和行为的时候,可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。

33、设计模式分类

	创建型	结构型	行为型	
类	factory method 工厂方法模式	adapter 适配器模式（类和对象）	template method 模板方法模式	interpreter 解释器模式
对象	abstract factory 抽象工厂模式 prototype 原型模式 singleton 单例模式 builder 构建器模式	bridge 桥接模式 composite 组合模式 decorator 装饰模式 facade 外观模式 flyweight 享元模式 proxy 代理模式	chain of responsibility 职责链模式 command 命令模式 iterator 迭代器模式 mediator 中介者模式	memento 备忘录模式 observer 观察者模式 state 状态模式 strategy 策略模式 visitor 访问者模式

34、创建型设计模式应用场景

设计模式名称	简要说明	速记关键字
Factory Method 工厂方法模式	定义一个创建对象的接口，但由子类决定需要实例化哪一个类。工厂方法使得子类实例化的过程推迟	动态生产对象
Abstract Factory 抽象工厂模式	提供一个接口，可以创建一系列相关或相互依赖的对象，而无需指定它们具体的类	生产成系列对象
Builder 构建器模式	将一个复杂类的表示与其构造相分离，使得相同的构建过程能够得出不同的表示	复杂对象构造
Prototype 原型模式	用原型实例指定创建对象的类型，并且通过拷贝这个原型来创建新的对象	克隆对象

Singleton 单例模式	保证一个类只有一个实例，并提供一个访问它的全局访问点	单实例

35、结构型设计模式应用场景

设计模式名称	简要说明	速记关键字
Adapter 适配器模式	将一个类的接口转换成用户希望得到的另一种接口。它使原本不相容的接口得以协同工作	转换接口
Bridge 桥接模式	将类的抽象部分和它的实现部分分离开来，使它们可以独立地变化	继承树拆分
Composite 组合模式	将对象组合成树型结构以表示“整体-部分”的层次结构，使得用户对单个对象和组合对象的使用具有一致性	树形目录结构
Decorator 装饰模式	动态地给一个对象添加一些额外的职责。它提供了用子类扩展功能的一个灵活的替代，比派生一个子类更加灵活	动态附加职责
Facade 外观模式	定义一个高层接口，为子系统中的一组接口提供一个一致的外观，从而简化了该子系统的使用	对外统一接口
Flyweight 享元模式	提供支持大量细粒度对象共享的有效方法	汉字编码
Proxy 代理模式	为其他对象提供一种代理以控制这个对象的访问	快捷方式

36、行为型设计模式应用场景（1）

设计模式名称	简要说明	速记关键字
Memento 备忘录模式	在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态，从而可以在以后将该对象恢复到原先保存的状态	游戏存档
Observer 观察者模式	定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并自动更新	联动
State	允许一个对象在其内部状态改变时改变它的行为	状态变成类

状态模式		
Strategy 策略模式	定义一系列算法，把它们一个个封装起来，并且使它们之间可互相替换，从而让算法可以独立于使用它的用户而变化	多方案切换
Template Method 模板方法模式	定义一个操作中的算法骨架，而将一些步骤延迟到子类中，使得子类可以不改变一个算法的结构即可重新定义算法的某些特定步骤	框架
Visitor 访问者模式	表示一个作用于某对象结构中的各元素的操作，使得在不改变各元素的类的前提下定义作用于这些元素的新操作	数据与操作分离

37、行为型设计模式应用场景（2）

设计模式名称	简要说明	速记关键字
Chain of Responsibility 职责链模式	通过给多个对象处理请求的机会，减少请求的发送者与接收者之间的耦合。将接收对象链接起来，在链中传递请求，直到有一个对象处理这个请求	传递职责
Command 命令模式	将一个请求封装为一个对象，从而可用不同的请求对客户进行参数化，将请求排队或记录请求日志，支持可撤销的操作	日志记录，可撤销
Interpreter 解释器模式	给定一种语言，定义它的文法表示，并定义一个解释器，该解释器用来根据文法表示来解释语言中的句子	虚拟机的机制
Iterator 迭代器模式	提供一种方法来顺序访问一个聚合对象中的各个元素，而不需要暴露该对象的内部表示	数据集
Mediator 中介者模式	用一个中介对象来封装一系列的对象交互。它使各对象不需要显式地相互调用，从而达到低耦合，还可以独立地改变对象间的交互	不直接引用

38、二叉树的特性

在二叉树的第 i 层上最多有2^{i-1}个结点（i≥1）。

深度为 k 的二叉树最多有2^k-1 个结点（k≥1）。

对任何一棵二叉树，如果其叶子结点数为 n0，度为 2 的结点数为 n2，则 n0=n2+1。

对一棵有 n 个结点的完全二叉树的结点按层序编号，即从第 1 层到$\lfloor log_2 n \rfloor + 1$ 层，每层从左到右依次编号。

具有 N 个结点的二叉树形态数：

$$A[N] = \sum_{M=0}^{N-1} (A[M] * A[N-M-1])$$

39、特殊的二叉树

满二叉树：任何结点，或者是树叶，或者恰有两棵非空子树。

完全二叉树：最多只有最小面的两层结点的度可以小于 2，并且最下面一层的结点全都集中在该层左侧的若干位置。

平衡二叉树：树中任一结点的左右子树高度之差不超过 1。

查找二叉树：又称之为排序二叉树。任一结点的权值，大于其左孩子结点，小于其右孩子结点。中序遍历结果有序。

40、最优二叉树的概念

最优二叉树：又称为哈弗曼树，它是一类带权路径长度最短的树。

路径是从树中一个结点到另一个结点之间的通路，路径上的分支数目称为路径长度。

树的路径长度是从树根到每一个叶子之间的路径长度之和。结点的带权路径长度为从该结点到树根之间的路径长度与该结点权值的乘积。

树的带权路径长度为树中所有叶子结点的带权路径长度之和。

41、图的遍历特点

深度优先遍历：

当以邻接矩阵作为存储结构时，深度优先搜索遍历图的时间复杂度为 $O(n^2)$

当以邻接表作为存储结构时，深度优先搜索遍历图的时间复杂度为 O(n+e)

广度优先遍历和深度优先搜索遍历图的运算时间复杂度相同，其不同之处仅仅在于对顶点的访问次序不同。

42、最小生成树与最短路径

（1）最小生成树解决方案

普里姆算法：找最短的边，直到把所有点连起来。注意：不能形成闭环。【贪心策略】

迪杰斯特拉算法：每一次只考虑从上一层节点到当前结点的最短路径。【贪心策略】

（2）最短路径问题解决方案

克鲁斯卡尔算法：从某个点开始，找现有点集合中最短的边。注意：不能形成闭环。【贪心策略】

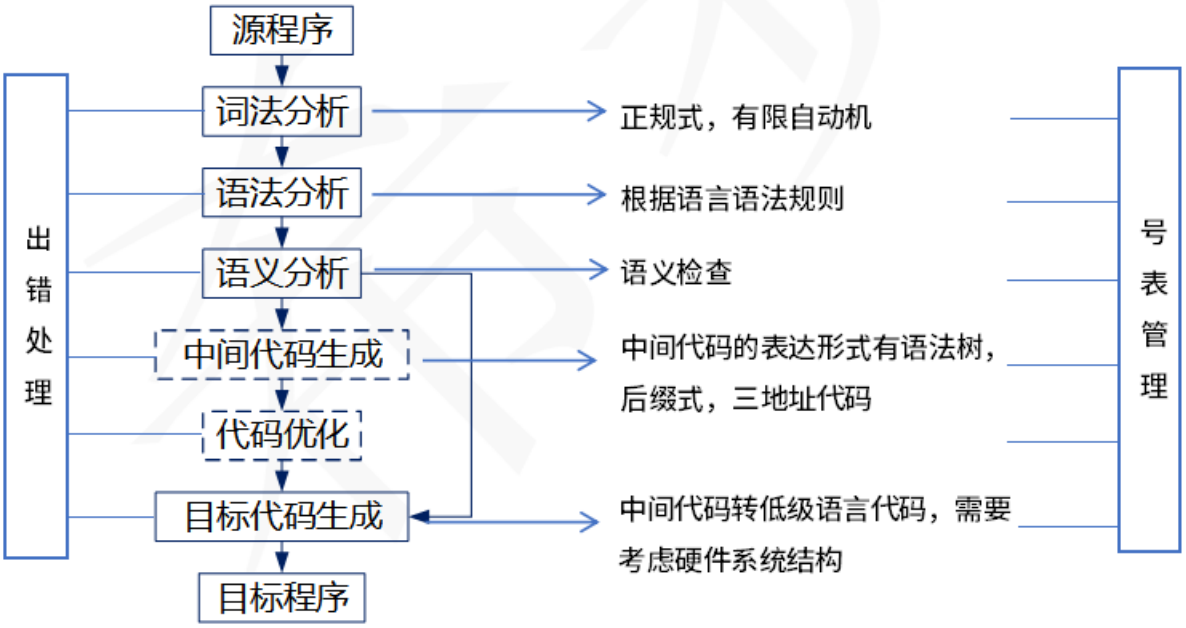
43、常见的算法执行所需时间的度量

$O(1)<O(\log_2 n)<O(n)<O(n\log_2 n)<O(n^2)<O(n^3)<O(2^n)$

44、常见排序算法对比

类别	排序方法	时间复杂度		空间复杂度	稳 定 性
		平均情况	特殊情况	辅助存储	
插入排序	直接插入	O（n^2）	基本有序最优 O（n）	O（1）	稳定
	Shell 排序	O（$n^{1.3}$）	-	O（1）	不稳定
选择排序	直接选择	O（n^2）	-	O（1）	不稳定
	堆排序	O（$n\log_2 n$）	-	O（1）	不稳定
交换排序	冒泡排序	O（n^2）	基本有序最优 O（n）	O（1）	稳定
	快速排序	O（$n\log_2 n$）	基本有序最差（n^2）	O（1）	不稳定
归并排序		O（$n\log_2 n$）	--	O（n）	稳定
基数排序		O（d（n+rd））	--	O（rd）	稳定

45、编译过程



46、文法和正规式

一般的程序设计语言属于上下文无关文法。

正规文法，表示的语言集合是正规集，正规集的规律可以用正规式表示。

正规式	正规集	举例
ab	字符串 ab 构成的集合	{ab}

a|b	字符串 a、b 构成的集合	{a , b }
a*	由 0 或多个 a 构成的字符串集合	{空 , a , aa , aaa , a…a(n 个 a)}
(a|b)*	所有字符 a 和 b 构成的串的集合	{空 , a , b , ab , aab , abb , baa, aba , …}
a(a|b)*	以 a 为首字符的 a、b 字符串的集合	{a , aa , ab , aab, aba , aaab , aaba , …}
(a|b)*abb	以 abb 结尾的 a、b 字符串的集合	{abb, aabb, babb, abaabb, abaabb, …}

47、保护范围和保护对象

法律法规名称	保护对象及范围	注意事项
著作权法	著作权 文学、绘画、摄影等作品	1、不需要申请，作品完成即开始保护 2、绘画或摄影作品原件出售（赠予）著作权还归原作者，原件拥有者有：所有权、展览权。
软件著作权法 计算机软件保护条例	软件著作权 软件作品	1、不需要申请，作品完成即开始保护 2、登记制度便于举证
专利法	专利权	需要申请，专利权有效期是从申请日开始计算
商标法	商标权	需要申请，核准之日起商标受保护
反不正当竞争法	商业秘密权	1、商业秘密包括技术与经营两个方面 2、必须有保密措施才能认定商业秘密

48、知识产权人确定-职务作品判定

情况说明		判断说明	归属
作品	职务作品	利用单位的物质技术条件进行创作，并由单位承担责任的	除署名权外其他著作权归单位
		有合同约定，其著作权属于单位	除署名权外其他著作权归单位
		其他	作者拥有著作权，单位有权在业务范围内

			优先使用
软件	职务作品	属于本职工作中明确规定的开发目标	单位享有著作权
		属于从事本职工作活动的结果	单位享有著作权
		使用了单位资金、专用设备、未公开的信息等物质、技术条件，并由单位或组织承担责任的软件	单位享有著作权
专利权	职务作品	本职工作中作出的发明创造	单位享有专利
		履行本单位交付的本职工作之外的任务所作出的发明创造	单位享有专利
		离职、退休或调动工作后 1 年内，与原单位工作相关	单位享有专利

49、侵权判断的特殊要求

中国公民、法人或者其他组织的作品，不论是否发表，都享有著作权。

开发软件所用的思想、处理过程、操作方法或者数学概念不受保护。

著作权法不适用于下列情形：

- 法律、法规，国家机关的决议、决定、命令和其他具有立法、行政、司法性质的文件，及其官方正式译文；
- 时事新闻；
- 历法、通用数表、通用表格和公式。

50、图的概念

（1）完全图

在无向图中，若每对顶点之间都有一条边相连，则称该图为完全图（complete graph）。

在有向图中，若每对顶点之间都有二条有向边相互连接，则称该图为完全图。

（2）强连通图：在有向图中，对于每一对顶点，从顶点 vi 到顶点 vj 和从顶点 vj 到顶点 vi 都存在路径，则称为强连通图。

该资料制作于 23 年 12 月，适用于第五版教材