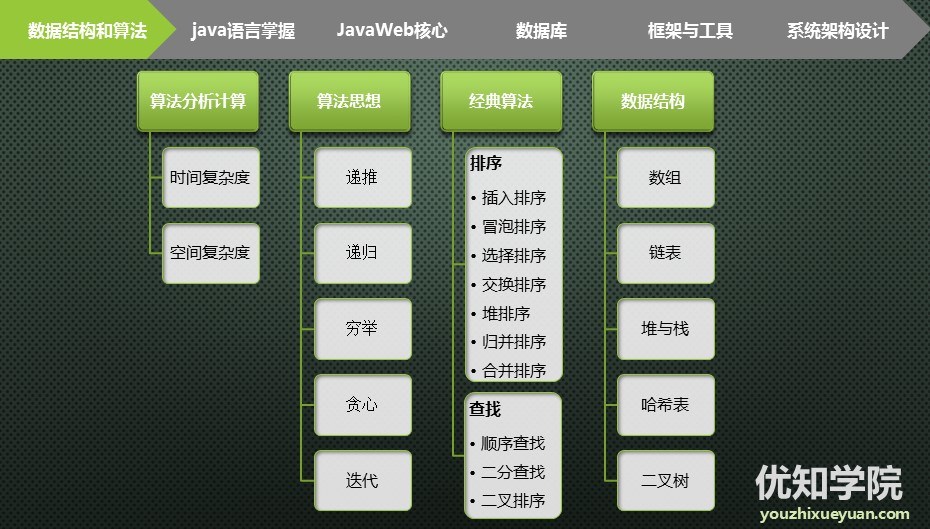
**编程基础：数据结构和算法**



**1 算法分析与计算**

算法时间复杂度和空间复杂度的分析计算

**2.算法思想**

递推、

递推是按照一定的规律来计算序列中的每个项，通常是通过计算前面的一些项来得出序列中的指定项的值。其思想是把一个复杂的庞大的计算过程转化为简单过程的多次重复，该算法利用了计算机速度快和不知疲倦的机器特点。

递归、

 一个过程或[函数](https://baike.baidu.com/item/%E5%87%BD%E6%95%B0)在其定义或说明中有直接或间接调用自身的一种方法，它通常把一个大型复杂的问题层层转化为一个与原问题相似的规模较小的问题来求解，递归策略只需少量的程序就可描述出解题过程所需要的多次重复计算，大大地减少了程序的代码量。

穷举、

穷举法的基本思想是根据题目的部分条件确定答案的大致范围，并在此范围内对所有可能的情况逐一验证，直到全部情况验证完毕。若某个情况验证符合题目的全部条件，则为本问题的一个解；若全部情况验证后都不符合题目的全部条件，则本题无解。穷举法也称为枚举法。

贪心、https://baike.baidu.com/item/%E8%B4%AA%E5%BF%83%E7%AE%97%E6%B3%95

贪心算法（又称贪婪算法）是指，在对[问题求解](https://baike.baidu.com/item/%E9%97%AE%E9%A2%98%E6%B1%82%E8%A7%A3/6693186)时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，他所做出的是在某种意义上的局部[最优解](https://baike.baidu.com/item/%E6%9C%80%E4%BC%98%E8%A7%A3/5208902)。

贪心算法不是对所有问题都能得到整体最优解，关键是贪心策略的选择，选择的贪心策略必须具备无后效性，即某个状态以前的过程不会影响以后的状态，只与当前状态有关

贪心选择是指所求问题的整体最优解可以通过一系列局部最优的选择，即贪心选择来达到

也是贪心算法与动态规划算法的主要区别

贪心选择是采用从顶向下、以迭代的方法做出相继选择，每做一次贪心选择就将所求问题简化为一个规模更小的子问题。对于一个具体问题，要确定它是否具有贪心选择的性质，我们必须证明每一步所作的贪心选择最终能得到问题的最优解。

当一个问题的最优解包含其子问题的最优解时，称此问题具有最优子结构性质

问题的最优子结构性质是该问题可用贪心算法或动态规划算法求解的关键特征

贪心算法对每个子问题的解决方案都做出选择，不能回退；动态规划则会根据以前的选择结果对当前进行选择，有回退功能。动态规划主要运用于二维或三维问题，而贪心一般是一维问题

分治、

动态规划、

把多阶段过程转化为一系列单阶段问题，利用各阶段之间的关系，逐个求解，创立了解决这类过程优化问题的新方法——动态规划

动态规划一般可分为线性动规，区域动规，树形动规，背包动规四类

迭代、

迭代法也称辗转法，是一种不断用变量的旧值递推新值的过程，跟迭代法相对应的是直接法(或者称为一次解法)，即一次性解决问题。迭代算法是用计算机解决问题的一种基本方法，它利用计算机运算速度快、适合做重复性操作的特点，让计算机对一组指令(或一定步骤)进行重复执行，在每次执行这组指令(或这些步骤)时，都从变量的原值推出它的一个新值，迭代法又分为精确迭代和近似迭代。比较典型的迭代法如“二分法”和"牛顿迭代法”属于近似迭代法。

分枝界限

[分枝定界](https://baike.baidu.com/item/%E5%88%86%E6%9E%9D%E5%AE%9A%E7%95%8C/10340480)法是一个用途十分广泛的算法，运用这种算法的技巧性很强，不同类型的问题解法也各不相同。[分支定界法](https://baike.baidu.com/item/%E5%88%86%E6%94%AF%E5%AE%9A%E7%95%8C%E6%B3%95/9902038)的基本思想是对有约束条件的最优化问题的所有[可行解](https://baike.baidu.com/item/%E5%8F%AF%E8%A1%8C%E8%A7%A3/962143)（数目有限）空间进行搜索。

**3常用数据结构**

数组、

链表、

堆、

栈、

队列、

Hash表、

二叉树等

**4经典算法**

**排序**

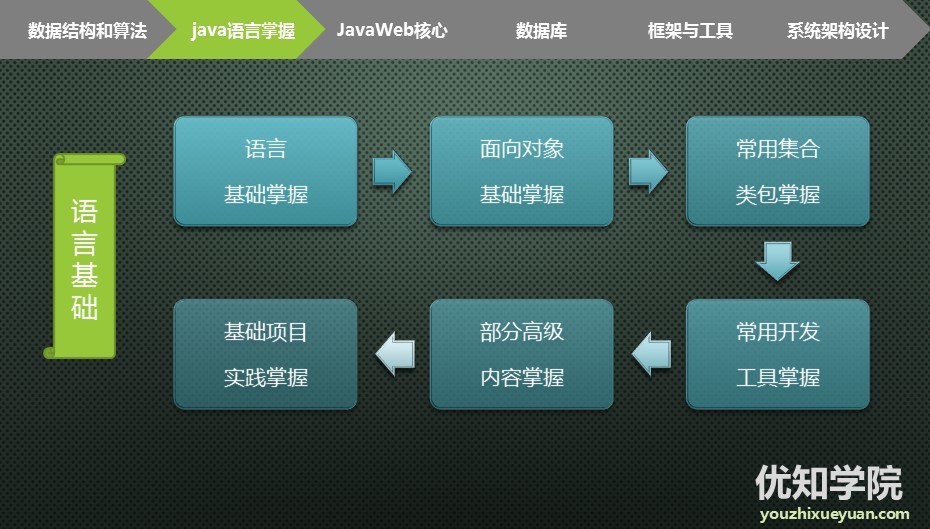
经典排序：插入排序、冒泡排序、快排（分划交换排序）、直接选择排序、堆排序、合并排序等

**查找**

经典查找：顺序查找、二分查找、二叉排序树查找

## ****java语言掌握****

**java语言基础**

**1语言基础**

Java语法格式，常量和变量，变量的作用域，方法和方法的重载，运算符，程序流程控制，各种基本数据类型及包装类

**2面向对象基础**

面向对象编程思想，类与对象，成员变量和局部变量，封装，this关键字，final关键字，static关键字。

**3 常用集合使用**

Collection以及各种List、Set、Queue、Map的实现以及集成关系，实现原理。

ArrayList，HashMap等常用集合优劣比较。

**Java语言高级**



**1.面向对象高级**

对象的三大特性：封装、继承和多态，优缺点

如何设计类，类的设计原则。

构造函数，内部类，抽象类，接口，对象的多态性，接口和抽象类的区别。

**2.异常处理**

Throwable/Error/Exception，Checked Exception vs. Unchecked Exception，异常的捕捉和抛出，异常捕捉的原则，finally的使用

**3.多线程**

创建与启动

线程和进程的概念

如何在程序中创建多线程，线程安全问题，线程之间的通讯

线程的同步与锁

死锁问题的剖析

线程生命周期

线程池

**4.输入与输出**

java.io包，理解IO体系的基于管道模型的设计思路以及常用IO类的特性和使用场合。

File及相关类，字节流InputStream和OutputStream，字符流Reader和Writer，以及相应缓冲流和管道流，字节和字符的转化流，包装流，以及常用包装类使用

分析IO性能

**5.反射**

类加载机制原理

反射构造方法、字段、方法

Properties配置文件

**代理、泛型、枚举、Java正则表达式等**

**6.网络编程**

网络机制

Socket原理机制

UDP、TCP传输等

**7.JVM深入理解**

一定要深入理解JVM原理，JVM内存划分、class加载机制以及GC策略等。

内存划分，Young Generation（年轻代）、Old Generation（年老代）以及Perm Generation（永久代）。

## ****java web 核心技术****



**1.前段技术**

html、css语法

css需要学习原生态，对css继承等掌握

js原生语法，js原生继承等的掌握

jquery

bootstrap

**2.Java Web**

建议把java web从容器启动到request、filter、listener了解原理，最好的方法就是通过调试代码一层层断点进入了解源码。

**3.模板引擎**

常见的模板引擎的语法掌握以及源码查看

**4.其他**

高性能

安全

事务JTA

其他需要了解的，如：管理JMX、安全JCCA/JAAS、集成JCA、通信JNDI/JMS/JavaMain/JAF、SSI技术。