算法：

为了充分利用计算机有限的资源，而提出的一个或者一系列解决问题的方案

时间复杂度：选择排序的时间复杂度 O(n^2)

空间复杂度：程序运行所需要的内存大小

栗子：

Int 21亿 4个字节 80亿字节 8000kb 8000M 8G

20亿 给定一个数，找是否存在 4g 分2批读

不存在的数有1个亿 400M

打标记 0和1来标记一个数是否存在 2g

标记不存在的数 100M

大文件的读取 10g 找出 前10个大的数 8g O(n)

A 5g 10 B 5g 10

顺序表： 一大块连续空间 增删慢 查找快

单链表：不连续的空间 每个节点记录下下一个节点的地址

双链表：每个节点记录下下一个节点的地址和上一个节点的地址

循环链表：链表尾节点记录首节点的地址，形成一个环

栈与队列：

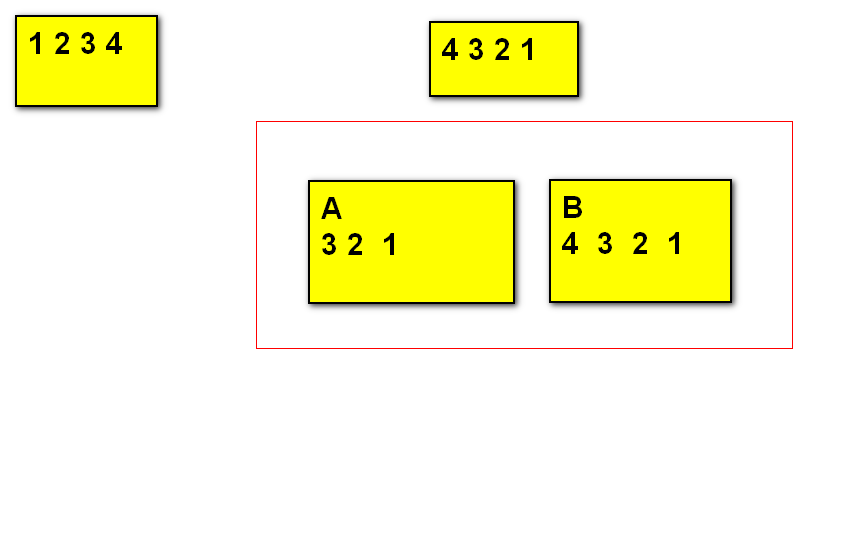
栈：先进后出 FILO

队列：先进先出 FIFO

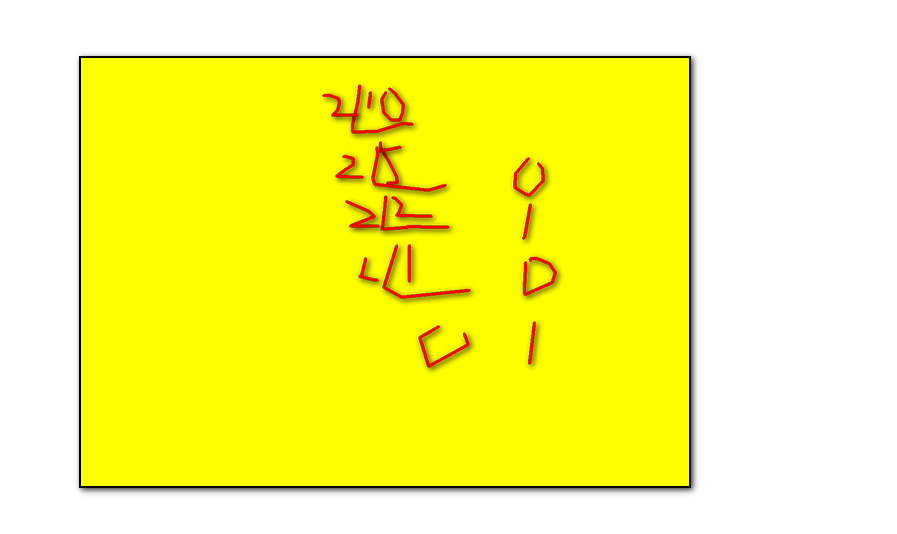
用两个栈实现一个队列？

A B两个栈倒元素

用两个队列实现一个栈？



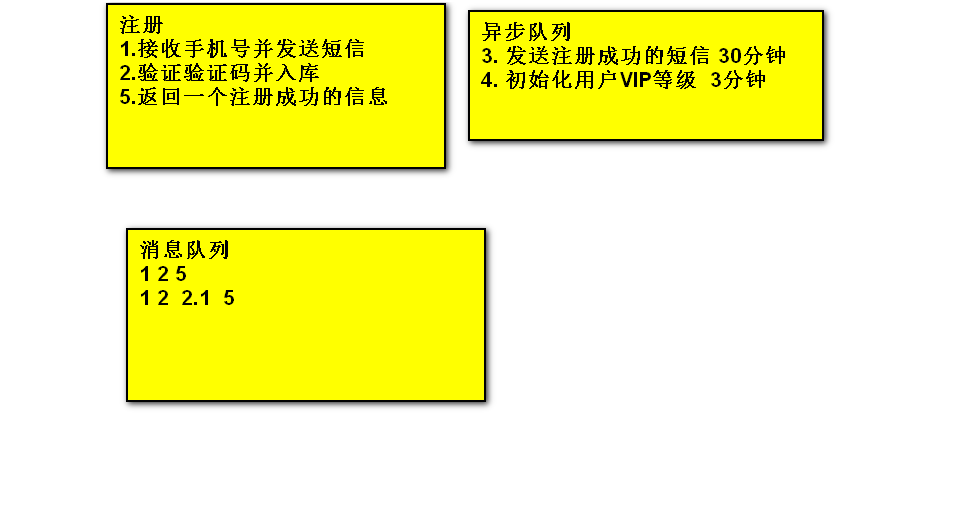
使用栈做进制转换 10进制 转2进制 7进制 转3进制

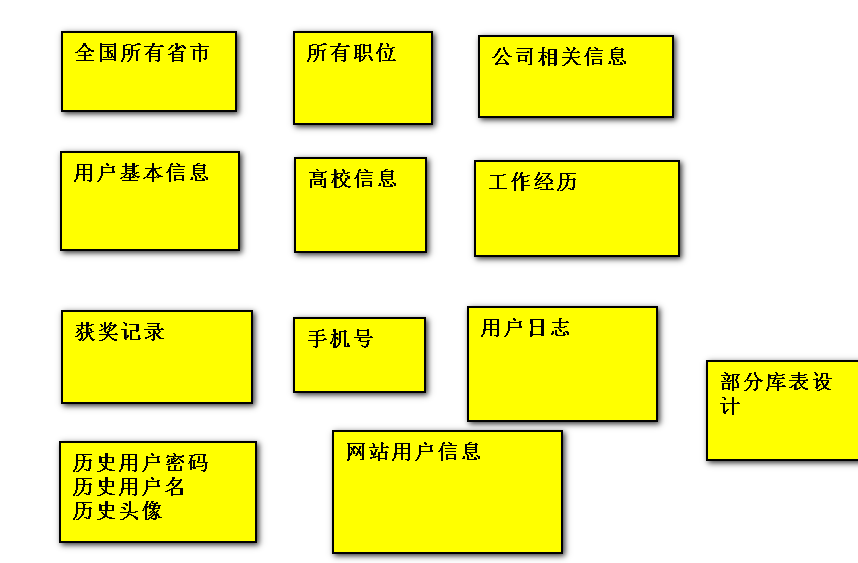


队列做异步处理：

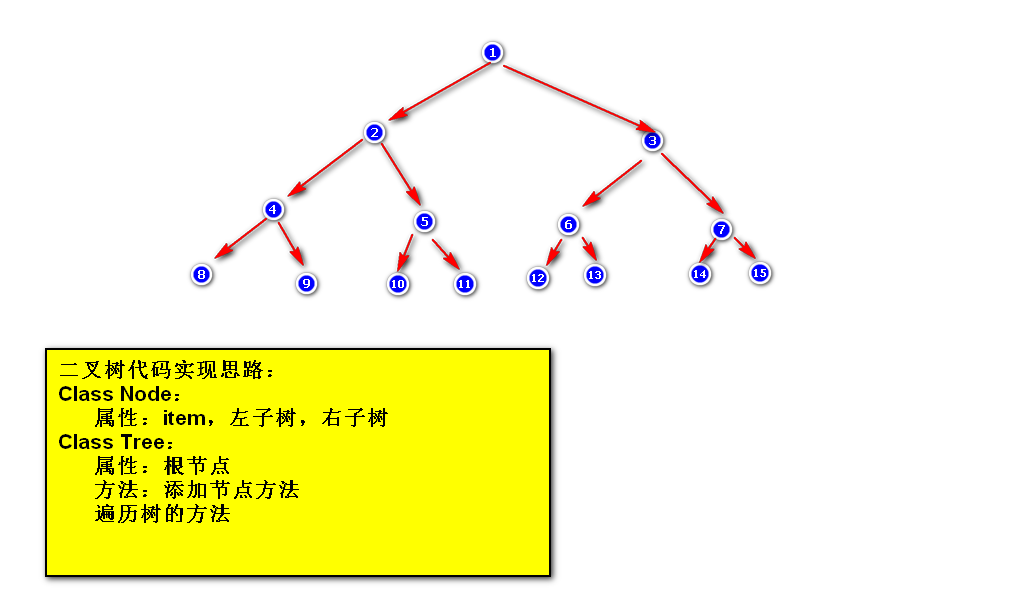
只考虑最终一致性的时候可以用异步队列来处理一些业务，以提高响应效率

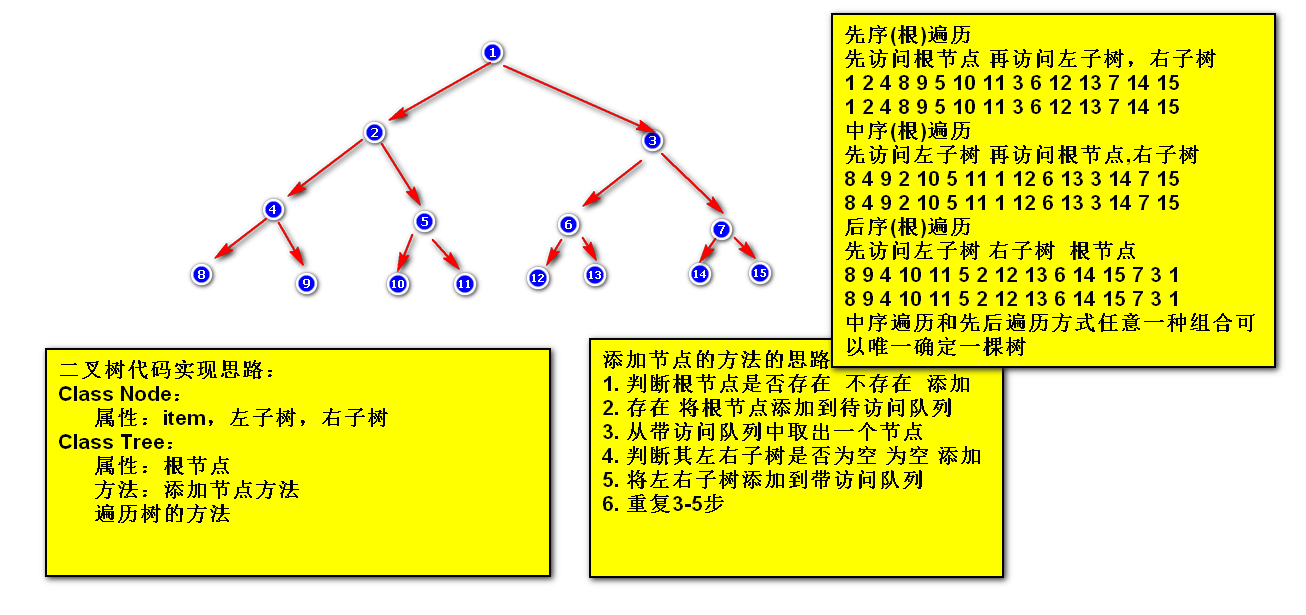
消息队列 celery 微服务





树

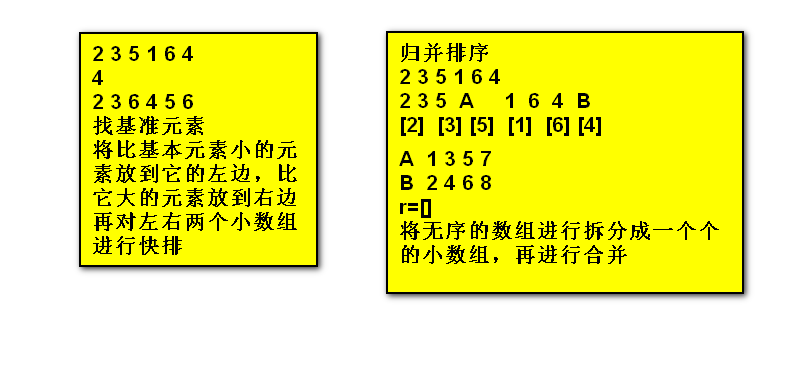




广度优先 深度优先 通用爬虫

二叉树 二叉查找树 B+树 B-树 AVL树 T树 红黑树 二分查找

排序：冒泡 选择 快排 归并 插入排序 希尔排序 堆排序 基数排序



时间复杂度的大小关系：

O(1) <O(logn)<O(n)<O(nlogn)<O(n^2)<O(n^3)<O(n^n)

堆排序

大根堆：树的叶子节点小于根节点 取出整棵树的根节点 再调整树 再取出…

小根堆：树的叶子节点大于根节点 取出整棵树的根节点 再调整树 再取出

插入排序 O(nlog(n))

最好：O(n)

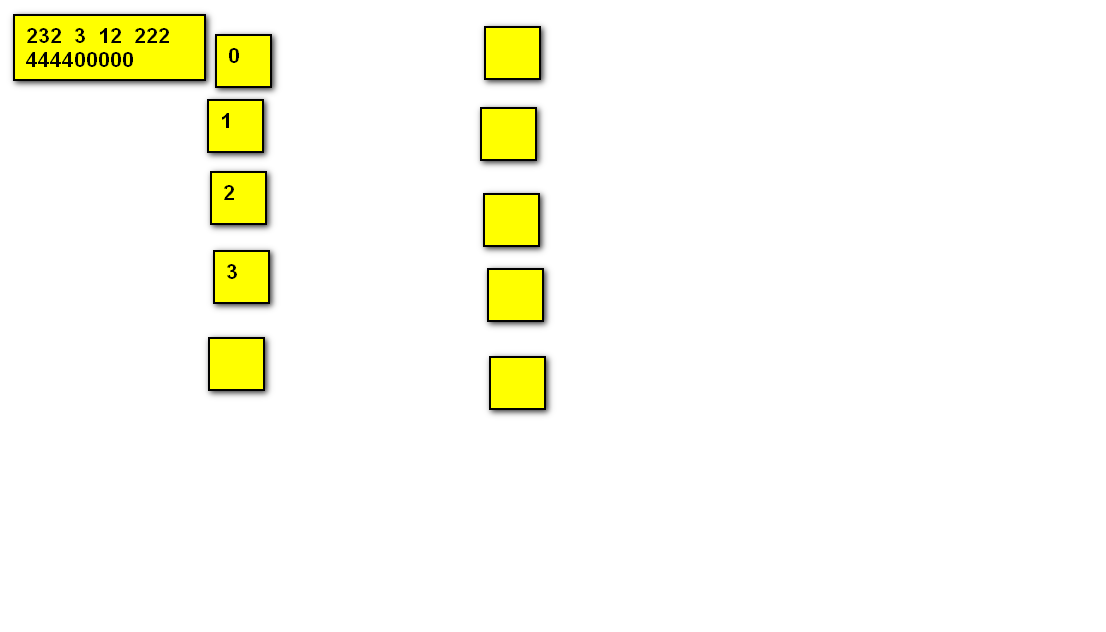
最坏：O(n^2)

在数组有序的情况下 在插入新数据时，只需要从头一个个的比较，直到找到比它大或者跟它相等的元素，就插入到这里

希尔排序：

希尔排序：希尔排序是把记录按下标的一定增量分组，对每组使用直接插入排序算法排序；随着增量逐渐减少， 每组包含的关键词越来越多，当增量减至1时，整个文件恰被分成一组，算法便终止

基数排序：



排序算法的稳定性：

在排序走完以后，如果相同元素的位置没有发生变化，则算法稳定，否是，不稳定