剑指offer

offer3：找出数组中重复的数字

- 利用hashset辅助(set.contains())
- 交换(num[i]==nums[nums[i]])
- 二分+计数(count>mid-low+1)
- 先排序(nums[i]==nums[i-1])

offer4：二维数组中的查找

- 从右上角开始比较(while循环)

offer5:替换空格

- 调用String函数(s.replace(" ","%20"))
- 先计算结果字符串长度(char[])
- 用StringBuilder动态存储(append)
- 先计算结果字符串长度(StringBuffer+setCharAt(index--,c))

offer6:从尾到头打印链表

- 利用栈(while(!stack.isEmpty()))
- 递归实现(recurse(head.next))

offer7：重建二叉树(前序+中序)

- 递归

  前序找到根节点的值

  根据值找到中序中下标

  然后递归调用

offer8：中序遍历二叉树的下一个结点

- 根据中序遍历(左*根右*)

  通过判断当前结点(根)有没有右孩子

offer9：用两个栈实现队列

- stack1进行push，stack2进行pop

offer10：斐波那契数列及青蛙跳台问题

- 递归(F(n)=F(n-1)+F(n-2))
- 用循环实现递归(for循环)

offer11：旋转数组的最小数字

- 二分法查找+特殊情况顺序查找
- 如果找目标值的话(4种情况缩小查找范围)

offer12：矩阵中的路径

- 递归+回溯(两层for循环每一个元素都可作为起点)

offer13：机器人的运动范围

- 递归(左上角出发，向下或者向右即可)

offer14：剪绳子

- 动态规划自下而上(两层for循环)
- 贪婪算法+数学(尽可能多分成3)

offer15：二进制中1的个数

- 位运算(无符号右移原数与1做与运算)
- 位运算(1左移后与原数与运算)
- 位运算(原数减1与原数做与运算)

offer16：数值的整数次方

- 分类讨论底数与指数(double存储+boolean判断合法性)

offer17：打印1到最大的n位数

- 在字符串上模拟数字加法(char数组)
- 采用递归(makeNumber(char[] number,int digit,int index))

offer18：删除链表结点(O(1))与删除重复链表结点

1. 删除链表结点(O(1))

   传入结点指针(单指针，尾结点，非尾结点，单个单个结点)

   传入结点的值(双指针，pre.next = cur.next)

2. 删除重复链表结点

   双指针(preNode+curNode)+boolean needDelete

offer19：正则表达式匹配

- 模式中第二个字符是否为"*",可分两种情况(是与不是)

  模式中第二个字符为"*",可分两种情况(当前字符等与不等)

  模式中第二字符为"*"，且当前字符等，可分三种情况

  ((str,pattern),(str,pattern+2),(str+1,pattern+2))

offer20：表示数值的字符串

- 可归纳为(A.B E/e C)

  其中A与C为有符号整数

  B为无符号整数

  '.'作为小数点判断

offer21：调整数组使奇数位于偶数前面

- 头尾遍历

  头指针找偶数，尾指针找奇数，交换，直到相遇

- 快慢指针

  从头开始遍历，遇到奇数时，将该奇数插入到该奇数前面的偶数之前

offer22：链表中倒数第k个结点

- 利用栈
- 利用相距为k-1的快慢指针

offer23：链表中环的入口结点

- 快慢指针

  先确定有环，再确定环结点数，最后找环入口

- 快慢指针

  快走两步，慢走一步，第一次相遇，停下

  快指针回头，快慢同步走，第二次相遇，返回

offer24：反转链表

- 三个指针(pre,cur,next)
- 递归

offer25：合并两个排序的链表

- 递归
- 非递归(while循环)

offer26：树的子结构

- 递归(三个函数，两个递归，一个比较)

offer27：二叉树的镜像

- 递归实现一
- 递归实现二

offer28：对称的二叉树

- 递归实现

offer29：顺时针打印矩阵

- 一个while，四个if，四个for

offer30：包含min函数的栈

- 两个栈(stack_data+stack_min)

offer31：栈的压入与弹出序列匹配

- 新建一个栈模拟压入弹出

offer32：从上到下打印二叉树

- 不分行从上到下打印

  辅助队列+ArrayList
- 要分行从上到下打印

  辅助队列+ArrayList+计数nextcount

  辅助队列+ArrayList+for循环(queue.size())
- 之字形从上到下打印

  辅助队列+ArrayList+计数nextcount+层数level

  两个栈(一个存从左到右，一个存从右到左)

offer33：二叉搜索树的后序遍历序列

- 递归(左右孩子与根节点比较)

offer34：二叉树中和为某一值的路径

- 递归(前序遍历)

offer35：复杂链表的复制

- 原地复制

  1.复制节点

  2.设置sibling

  3.拆分长链表

offer36：二叉搜索树化双向链表

- 递归(中序遍历)

offer37：序列化二叉树

- 递归(前序遍历)

offer38：字符串的排列

- 递归(固定第一个，求后面字符排列)

offer39：数组中出现次数超过一半的数字

- 快排找到中位数后再判断
- 守卫士兵后判断

offer40：最小的k个数

- 快排
- 堆排序

offer41：数据流中的中位数

- 大顶堆+小顶堆

offer42：连续子数组的最大和

- 用sum+maxSum动态判断

offer43：从1到n整数中1出现的次数

- 数学归纳出规律

offer44：数字序列中某一位的数字

- 数学归纳出规律

offer45：把数组排成最小的数

- ArrayList+Collection.sort()

offer46：把数字翻译成字符串

- 递归+自下而上(解决重复子问题)

offer47：礼物的最大价值

- 动态规划+辅助一维数组

offer48：最长不含重复字符的子字符串

- 动态规划+maxLength+curLength+preLength

  以第i个字符为结尾的不含重复字符的子字符串的最大长度

  我们从第一个字符开始遍历，定义两个int变量preLength和curLength来分别代表f(i-1)和f(i)，再创建一个长度为26的pos数组来存放26个字母上次出现的位置

offer49：计算第n个丑数

- 从数组中已有的丑数里找到三个丑数T2、T3、T5，它们分别和2、3、5相乘得到的值恰好比已有的最大丑数大，三个乘积中最小的一个就是下一个丑数，存放入数组中，同时更新T2、T3、T5，使它们仍然保持与2、3、5的乘积恰好比已有的最大丑数大

offer50：字符串(或流)中第一个之出现一次的字符

- 输入为字符串

  哈希表+一遍扫描计数+一遍扫描找

- 输入为字符流

  哈希表+动态(一遍扫描计数+一遍扫描找)

offer51：数组中的逆序对

- 归并排序+动态计数

offer52：两个链表的第一个公共结点

- 利用长度关系
- 利用双指针(浪漫相遇)

offer53：在排序数组中查找数字

- 数字在排序数组中出现的次数

  改进的二分法((first+last)或(first+0.5,last+0.5))

- 0到n-1中缺失的数字

  改进的二分查找

- 数组中数值和下标相等的元素

  改进的二分查找

offer54：二叉搜索树的第k个结点

- 中序遍历+计数

offer55：二叉树的深度与平衡二叉树判断

- 二叉树的深度

  递归+Math.max()

- 平衡二叉树判断

  递归+剪枝

offer56：数组中数字出现的次数

- 数组中只出现一次的两个数字

  异或+分组

- 数组中唯一只出现一次的数字

  所有数字的二进制表示的对应位都加起来，如果某一位能被三整除，那么只出现一次的数字在该位为0；反之，为1

  两层for循环(第一层遍历数字，第二层遍历二进制码32次)

offer57：和为s的数字

- 和为s的两个数字

  头尾指针

- 和为s的连续正数序列

  small+big+curSum+ArrayList

offer58：翻转字符串

- 翻转单词顺序

  递归(翻转整个句子后逐个翻转单词)

- 左旋转字符串

  递归(先翻转前部分与后部分后翻转整个字符串)

offer59：队列的最大值

- 滑动窗口的最大值

  建立一个两端开口的队列，放置所有可能是最大值的数字（存放的其实是对应的下标），且最大值位于队列开头。

  从头到尾扫描数组(有四种情况)

- 队列中的最大值

  利用一个双端队列来存储当前队列里的最大值以及之后可能的最大值。

  双端队列data+maximum+存放数据与下标的结构体

offer60：n个骰子的点数

- 递归+(6*n-n+1)大小的数组
- 循环+两个数组来分别存放本轮循环与下一轮循环的各种点数和出现的次数

offer61：扑克牌中的顺子

- 排序+numberOf0+numberOfGap

offer62：圆圈中最后剩下的数字

- 链表存数据+对长度取余来循环
- 数学推导规律

offer63：股票的最大利润

- 遍历每一个数字，并保存之前最小的数字，两者差最大即为最大利润

offer64：限制条件1累加到n

- 逻辑运算符(代替if)+递归(代替循环)

offer65：不用加减乘除做加法

- 异或(相加二进制)+逻辑与(判断进位)

offer66：构建乘积数组

- 先计算并保存左半部分为B[i]，然后计算后半部分顺便把与之前保存的B[i]相乘可得到最终的B[i]

offer67：把字符串转换成整数

- 每个字符转化为数字，并进行累加即可(考虑情况多)

offer68：树中两个结点的最低公共祖先

- 二叉搜索树

  最低公共祖先结点的大小在两个树结点大小的中间

- 普通二叉树

  递归(遍历将子结点的信息往上传递,在左右子树中进行查找是否存在两个树结点)