制表

代码中文化 每个题看复杂度 正确性

看题：归纳函数名 变量名 关联和事件

一、1、程序设计就是解决问题 读入某些值 输出某些值

问题中输入大量变量，要求编程能对所有符合条件的变量都输出正确结果

然后还需要满足时间和空间的需求 多少秒 内存限制多大

乐队：邀请了l个乐队，之后还会要求更多，每天1队，今后n天可租,每天租金不同 求最低平均价格最低的租法 必须连续租

只用文字输入输出、时间内存限制、答案正确与否（各种输入值测试）、可以实时报告运算时间和内存（方便优化）

如何解决问题？ 编码和调试 时间复杂度 暴力解决（穷举） 分治 动态规划 线性数据结构 队列 栈和双端队列 树的实现和遍历（广搜层序） 二叉搜索树 优先级队列和堆 图的表现和定义 深搜广搜 最短路径

Acm top-coder:组件设计

**在线评分网站：oj poj**

团队：打草稿， 结构 要使用的api 函数名变量名 中文伪代码

任务安排，结对编程 先解决问题哪部分，用什么方法？

执行前条件（调用条件） 执行后条件（返回条件 ） 定义类

二、如何解决问题？ 把问题写在本子上，苦想，打草稿，在本子上写出答案

理解问题的 已知 未知 限制条件 制定计划写文档和伪代码草稿，查找列举工具 寻找类似的模板 用自己熟悉的语言（术语）重新归纳问题

仔细阅读题目、需求： 要求理解无错，了解每个条件边界 出现的**所有数据**

抽象化：把问题 转化为数学，程序的模型 比如对象化 或者程序化 把数据转换为程序数据 概念转换为程序流程

程序策略：用什么工具 什么api 什么思路 什么数据结构 什么算法

需要验证策略是否正确

回顾和优化： 把做过的题 和出现的错误都记载下来 思考更好的方法

2、如何解题： 靠直觉和记忆 （1）以前做过同样或者类似的题目吗？ 存在变形吗？ 虽然表面上不一样仍然是变形 （2）从最基本的解题方式入手，枚举所有可能性，依靠电脑性能

N个糖果<=30 分个三个孩子，糖重不同<20整数 尽可能公平，使最大最小重差距最少，求最小差距 糖最小为1 最大可能是20 输入各糖果的重量 ，列举所有分配情况，a/b/c 广搜 减去极限以外的不可能情况 减去不可能情况 与孩子顺序无关（/3）

可以写出问题公式吗？

能够简化问题， 选取其中一些特殊情况吗？（**比如取值最少的情况**） 归纳法，阶乘 从最简单形式找规律 二维简化为一维（x,y方向单独讨论） 无限多的值变为有限的几个值 分解条件，分成多个小问题

可以画成草图吗? 用图形关系表示 单轴区间图 十字坐标轴 关联图 函数图

可以反顺序解决问题吗？ 从结果到问题 从小到大

无序的问题可以强制排序，给问题排序 编号 顺序无关的问题加上顺序也不会影响

缩小答案范围： 答案明显是在某个范围内的

三、编码和调试

简洁 代码复用，出现三次的代码就定义为函数 一个函数一个功能 多使用标准库

字典关联数组

把基础功能包装以后用（比如交换数组元素）

使用明确的命名

代码和数据分离： switch(i) return a 表达式中尽量不要硬编码魔数，直接量，使用局部或者全局变量 有多个选项的选择应该用集合保存所有状态 数据，比如数组或者结构 多种输出可能，可以用集合保存所有输出值 多种参数可能，用结构保存 {1,2,3,4,5} [‘a’,’b’,’c’] 保存棋的坐标

3、常见错误 运算溢出

超出数组范围 访问 赋值

集合范围：开区间 闭区间 半开区间 下标从0到n-1 迭代器从begin end python切片范围 js splice substring范围 使用不同的范围表示n也不同

差一个错误 多一个 少一个 跟比较号有关< <= 导致循环多一次

栈溢出：递归太多

多维数组的下标乱序

比较函数错误： 反自反性 a<a 为假 非对称性 a<b则b<a为假 传递性a<b<c a<c

等价传递性 a==b==c **a不小于b且b不小于a则ab相等 a<b为假且b<a为假**

最小值和最大值是例外： 素数中 0 1不是素数 2是偶数也是素数

运算符优先级错误

效率错误：使用错误的输入输出方式，使用错误的比较方式（字符串比较）

变量未初始化：是否有默认值？是多少？ 没有默认值会报错？还是会是错误值？

调试和测试：单步 利用少量的输入测试 使用断言 false报错 测试：输入值对比预期输出值 然后测试 自动化输入输出测试 使用输入表 输出表（两个数组）

4、变量取值

运算溢出 过大的结果 加法乘法 过大的中间值 过小的负值

无穷大值：2-32

避免溢出：

数据类型提升： 整数转实数 int转long 有符转无符 小于short的相加 **字面量默认类型**，

颠倒运算顺序

最小公倍数

二项式系数的 递归关系式

5、实数运算 精度 范围 x\*1.0/x ==1? 电脑的实数是近似的，计算过程会发生各种变化 很可能运算后反运算结果不同

ieee

四、算法复杂度分析

五、算法正确性

输入输出单元测试不能保证算法正确

1、数学归纳法：分阶段 第一阶段证明 归纳证明，下一阶段也成立

2、循环不变式 算法中一般带循环 每次循环都检查是否满足一个**固定条件** i<0 i++ 循环过程始终满足不变式 终止时得出答案 if() while() 终止时 计数器 状态变量的值 for() 语句中，i++是本次循环结束才执行 i<n是本次循环开始前执行

循环停止，说明此时已经不满足条件 而最后一次循环就没执行 ，i<10 只会执行0-9，共十次循环

不变式是循环继续条件 不满足的时候代表循环终止，也可以认为是循环终止条件

另一种写法，在循环内部判断，if() 用break 和continue来控制和跳出循环

3、归缪法 假设和期望相反的情况，分析得到假设错误的结论

假设存在更优解，证明更优解不存在

4、鸽笼原理 人多于位置 ，必然产生重复 鸽子多于笼子，必然一个笼子装几只

六百个人肯定有人生日重复

构造性证明：证明存在期望答案 列举答案

算法设计范式

六、暴力解决（穷举） 2、递归调用 穷举搜索

**把操作分解成小操作，执行其中一个，剩下的调用自己完成，范围会越来越小**

**常用在对集合的操作上，对越来越小的集合进行操作 set型和get型**

**常表现为集合中的 元素和小集合间的关系**

问题 子问题：相同的

七、分治 递归是分解成一个小问题和剩下的问题 分治是分成同等大小的几个子问题

分解divide 答案归并merge

八、动态规划：和 分治类似 但是子问题答案用于多个问题，需要保存 重复利用结果

缓存 **重复子问题** 需要**制表**，只算一次重复子问题 穷举

九、动态规划技巧

十、贪心法 局部最优解 分割子问题，每个问题选当前最优解

十一、组合搜索 穷举搜索

启发式函数 制表

十二、优化问题转决策问题（是否,true,false 是否存在？超过，大于，相等） 再使用二分法

十三、数值分析

十四、整数论

十五、计算几何

数据结构

十六、位掩码

十七、部分和

十八、线性数据结构 动态数组 链表

十九、队列，栈。双端队列

二十、字符串

二十一、树实现和遍历

二十二、二叉搜索树

树堆

二十三、优先队列 和堆

二十四、区间树

二十五、互斥集合 （并查集union-find,连通性问题）

二十六、字典树

二十七、图

二十八、深搜

二十九、广搜

三十、最短路径

三十一、最小生成树

三十二、网络流