

关于这篇课件

- 关于CSP常见考点的一些梳理
- (虽然我也只刷掉了从NOIP存在以来到2016年的提高组全部题目, AK过NOIP2018D1)
- 算法按照我认为的常考程度/重要程度排序。(本课件共34题)
- 希望个人的汇总能让大家看到一些隐藏在题目背后的普遍规律吧。
- 也希望我的微薄努力,能幻化成一点星光,照亮前路,让大家不再迷茫,一路勇往无前。
- 以梦为码,逐鹿天涯。
- 背景Pixiv71471901P3。

打表找规律

- 打表找规律,就是通过计算机的计算,求出一定范围内的答案,然后通过人类智慧找出答案之间的关系。
- 这里最经典的就是NOIP2017: 小凯的疑惑
- 答案就是a * b a b,暴力背包+打表很容易观察。
- ·然而我选择了手推exgcd,浪费了不少时间。
- 遇到这种题目不妨在手玩之前先让机器帮你玩一下。
- 不要让计算机下岗嘛。
- (单隐层神经网络在规模足够大的情况下可以以无限逼近精度拟合任何连续函数。你的大脑可比这东西复杂多了)

模拟

- 顾名思义,就是按照题意进行操作。利用计算机的算力,快速得出答案。
- ·不要小看模拟题,真的很重要。别人都AC的题你丢分了,那你基本上就能考虑回哪个班学文化课了。
- NOIP2016: 玩具谜题
- 直接按照方向模拟转圈即可。
- NOIP2017: 时间复杂度
- ·写一个解释器,计算时间复杂度? (这题我一个函数少打了return, 然后就爆零了·····)

模拟

- NOIP2003: 侦探推理
- 算法十分显然: 枚举今天是星期几, 然后枚举谁是罪犯。
- 判断是否有解,解是否唯一。
- 难度在于字符串处理!
- ·建议这种简单的字符串处理不要用传统字符串数组,用string类。
- 无论是空格还是换行符的处理都要简单得多。
- (我当时NOIP2017: 时间复杂度就是用传统字符串写的,特别麻烦,根本调不出来。当然也不排除是我太菜了)

模拟

- · NOIP2005: 等价表达式
- 思路十分显然,枚举几个数,带进去,看数值是否相等即可。
- ·如果是python的话,直接eval()即可。
- 然而 C + + 需要自己写多项式解析,维护两个栈即可。
- 这种东西一定要熟练!别人分分钟写完你浪费半天最后也会凉凉。



- · NOIP非常喜欢考动态规划,几乎是每年必考。
- · CSP估计也会这样吧。
- · NOIP阶段的DP题目一般都不是太难。
- 合理设计状态,列出转移方程,写出暴力,60分就到手了。
- 之后考虑数据结构/方程性质优化(决策单调性, 斜率优化)等。
- 如果实在想不到怎么优化,就暴力走人。不要浪费太多时间。
- 注意不要把一些看起来很像找规律的DP数学题当成找规律题做。 那样真的找不出来规律。

- 常见考点:
- 按照优化方式:
- \bullet 暴力DP,数据结构优化DP,斜率优化DP,决策单调性优化DP,矩阵 优化DP。
- 按照表现形式:
- · 序列DP,树上DP,图上DP,状态压缩DP,数位DP,组合计数。
- 按照实现方式:
- · 普通迭代DP,记忆化搜索DP,分治实现DP,动态DP。(注意记忆化和迭代在某些迭代难以实现的题目上有奇效)







- NOIP2006: 2^k进制数
- 先求出答案最多有多少个 2^k 进制位,记录为a,恰好有a位时最高位最大数字,记录为b,然后考虑DP。
- f[i][j]表示含i个2k进制位,最低位≤ j的数字个数。
- 显然这个数字内不能有任何一位为0。我们可以将每一位转化为2^k减去这一位的值。这样就变成了左边每一位比右边大。
- 每次向从高到低数的第一位添加数字。f[i][j] = f[i][j-1] + f[i-1][j-1]。
- 答案为 $\sum_{i=2}^{a-1} f[i][2^k 1] + \sum_{i=1}^b f[a-1][2^k 1 i]$
- 前面是枚举长度,后面是枚举最高位的值。
- 然后写一个高精度就好啦。



- NOIP2017: 宝藏
- 枚举每个点的父节点能拿到70分? 摸了摸了。
- 考虑状压暴搜,枚举一个起点,在搜索的过程中动态维护每个点的深度信息。对于一个状态,枚举已经在树中的一个点,加入一个没有在树中的点。只在更新后更优的情况下递归搜索。(复杂度? 反正不是n!)

- NOIP2017: 逛公园
- 先把距离为0的块缩起来。从1, n分别跑最短路。
- ·如果最短路经过了块,则答案为inf。
- 否则令f[i][j]表示从1开始到点i,多走长度为j,方案数。显然j不能大于50,且转移中j单调不减。
- 跑出最短路图,拓扑排序转移即可。
- •最后别忘了取模! (70变30的故事)



- NOIP2018: 保卫王国
- ·显然的DP, f[i][0/1]表示点i有/无军队, 子树合法的最小代价。
- $f[i][0] = \sum_{j|sons_i} f[j][1]$
- $f[i][1] = val[i] + \sum_{j|sons_i} \min\{f[j][0], f[j][1]\}$
- · 钦定一个点? 把转移写成矩阵形式,倍增维护一下就好。点i的结果矩阵为一个初始矩阵和子树结果矩阵的连乘积)
- ·不要写树剖+线段树维护动态DP,我当时写了没调出来(这有何 难.jpg)

- 贪心算法是指: 在当前的每个位置选择最优解, 一定能得出全局最优解。
- 和动态规划的区别为:动态规划的局部最优方案不一定包含于全局最优,而贪心算法的局部最优一定包含在全局最优当中。
- 贪心的常考程度与动态规划不相上下。
- 但是普遍代码难度小于动态规划?
- 考点:不可整理(依据题目性质分析解题策略)

- NOIP2018: 铺设道路
- 我们记录答案为更新完前 i 个点后的最优解。
- 考虑第i个点和第i 1个点的高度,如果第i个点更小,就在填第i 1个点的时候顺便把它填了,否则需要多填的次数就是两者的深度之差。
- · 这题考场上10分钟A不掉你就没时间扫雷了。

- NOIP2018: 赛道修建
- 最小值最大,一眼二分答案。(注意限制一下二分边界防止卡常数)
- · 考虑如何验证。我们令f[i]表示从i点上面向下,所能达到的最长长度。
- 如何计算在满足限制情况下最多能组成多少条赛道。
- •将点i的孩子的f[]按照放进一个set,能单独出来的提前删去。
- 然后从小到大单独试图为每一个进行配对,配对成功删去。
- · 将set中剩下的最大值作为f[i]。
- 正确性? 因为每个孩子的f最多用一次,所以在i这里先进行配对显然是不劣的。(这题30分钟A不掉的话,你依旧没时间扫雷)

- · NOIP2018: 旅行(从这里我们可以看出NOIP2018考了三道贪心)
- 图中最多有一个环。
- 将每个点的相邻点按照从小到大排序。
- 枚举断掉哪条边,从1开始暴搜即可。
- 正确性显然。
- 注意常数。
- · vector 存边+每次重构+每次 sort 的我被卡了12分(老年选手无所畏惧)

- · NOIP2008: 双栈排序
- 首先两个栈中的元素都应该是单调不增的。
- 对于每次操作,先判断能否弹栈。因为没有相同数,所以可以弹栈时一定要弹。
- •对于能压入a栈的数t,找到它后面第一个同时比它和b栈栈顶大的数c。
- 在c后面,不能有比t更大的数f。
- ·因为c如果压入a栈,则必须弹出t,f会把t卡住。
- 而c如果压入b栈,当前b栈栈顶≥ a栈栈顶,所以也能用t判断



图论

- 主要考点:
- ·图的连通性(tarjan: 点双、边双、强连通)
- ·最短路(SPFA已经死了)。
- 生成树(以及计数)
- 拓扑排序
- ·网络流:最大流Dinic,最小费用最大流。(会打板子即可)
- · 费用流有时候对你懒得思考的DP题有奇效。

图论

- · NOIP2013: 华容道
- 棋盘上除了关键棋子和空白格子以外所有东西都是无差别的。
- 定义d[i][j][0...3][0...3]表示关键棋子在(i,j),空白格子相对关键棋子的位置,关键棋子要去的位置相对关键棋子的位置。
- · bfs预处理一下。
- · 每次询问,先把空白格子移动到关键棋子旁边,然后spfa。
- 这种题目考场上建议写暴力吧。



- · NOIP2015: 运输计划
- 最大值最小,一眼二分答案。
- 如何验证? 考虑所有长度超过当前答案的路径,它们的交集上面一定有一条路径需要被改造。
- 且被改造的路径长度应大于它们超出长度的最大值。
- 这题做完了。



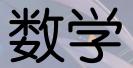


图论

- NOIP2000: 方格取数
- · k取方格数? 费用流板子题。
- ·每个点拆成两个,连费用正,流量1的边,费用0,流量inf的边。
- 跑最大费用流即可。
- NOIP2008: 传纸条
- 和上一题思路相同点拆成两个,只连费用为输入值,流量1的边。
- 跑最大费用流。

数学

- · CSP范围内的数学题知识有:
- •取模,同余,逆元,exgcd,中国剩余定理,合并同余方程
- · 素数判定(根号, miller rabin), 线性筛法
- ·基础组合数学(组合数,卡特兰数,第一、二类斯特林数,斯特林反 演),置换群,染色,Polya定理,Burnside引理等。
- •数论函数 (μ , φ ,id)等,基础的莫比乌斯反演
- 期望,线性方程组,高斯消元
- 进制转化,高精度,FFT, NTT, FWT (及FWT的构造)等
- · 类欧几里得算法(秦神称之为「真·欧几里得」)
- 划掉的部分可以直接选择暴力滚粗了2333



- NOIP2011: 聪明的质监员
- ·显然Y随w增大单调不增。
- •所以|Y S|随w增大先减后增。
- ·大力三分w即可。
- •二分w,求出使得 $Y \le S$ 的最大的w,然后带入w和w + 1进行运算取min。





- NOIP2018: 填数游戏
- 考虑找规律。
- n=1时,显然答案为 2^m 。
- n = 2时,答案为 $4 * 3^{m-1}$ 。
- n = 3时,答案为 $112 * 3^{m-3}$ 。
- 65分到手。之后你可以继续找,也可以弃疗了……
- (这题当年难倒集训队2333)

数据结构

- ·数据结构是NOIP中的经典考题。
- 主要锻炼选手的代码能力(并满足出题人毒瘤欲望)
- 需要熟练掌握的数据结构有:
- 二叉堆, 左偏树;
- · 线段树(动态开点/可持久化)、树状数组、轻量级平衡树(非旋转treap, 替罪羊树),重量级平衡树(Splay),LCT, ETT;
- · 根号数据结构: KD Tree, 可持久化数组。
- · 单调队列,对顶堆,pbds等科技。
- ·数据结构题和模拟题一样,如果别人都AC了就你挂分了,那你凉凉了。

数据结构

- NOIP2016: 蚯蚓
- •一个变量维护增加值。
- 初始的降序排序,开三个队列,维护初始的,第一段,第二段。
- 显然先切的分成两段也比后切的分成的两段长。
- 所以三个单调队列就搞定了。
- · 想想当年多少写二叉堆暴力的铁憨憨,其实直接phds配对堆就能……



- NOIP2017: 列队
- 动态开点线段树或平衡树大力维护每一行和最后一列。
- 线段树/平衡树中每个节点代表一段区间。
- · (别说了,我当年就是个Splay没调出来的菜狗。这有何难.jpg)

数据结构

- · NOIP2016: 天天爱跑步
- 这题有非数据结构的做法,但是这里我想讲一个利用数据结构的暴力做法。
- 路径可以拆成上行/下行两段。
- •上行路径上每个人到路径上的点i的时间可看作 $t_0 dep[i]$ 。
- 下行路径上每个人到路径上的点i的时间可看作 $t_1 + dep[i]$ 。
- 而观察员观察的时间,可看作a dep[i]或b + dep[i]。
- 按照加减分类,按照常数排序,转换为树链加,单点查问题。
- (当然也可以不转换,对一条链进行操作时直接主席树区间加)
- (其实打标记,开一个桶,在遍历整棵树的时候维护一下就好了)

字符串

- · 字符串题目在NOIP中出现不多,相对难度不大。
- (不过今年不一定)
- ·怕是难度全在IO。
- 需要掌握的算法有:
- *KMP*(深入理解), *AC*自动机。
- 后缀数据结构: 后缀数组, 后缀自动机, 后缀平衡树, 后缀树。
- 为什么不试试神奇的哈希呢?

字符串

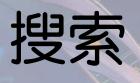
- NOIP2015: 子串
- 注意到相邻的两个字串不会被强行拼成一个。
- f[i][j][k]表示考虑利用a串前i位,b串前j位已经匹配,子串数量为k,方案数,sum[j][k]表示f[i][j][k]在i维度上的前缀和。
- 如果b[i] == a[j]的话,向前枚举本次匹配串的左端点,进行转移。
- ·发现维度i没什么用,滚动数组滚掉。
- · 最终答案即为sum[len(b)][input_k]。



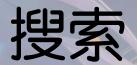
搜索

- 搜索是指:从已知状态的临域出发,寻找未访问过的新状态或更优状态的方法。
- ·常见的搜索方法有: dfs和bfs。
- 常用于图论题和动态规划题。
- ·以及各种暴力(万物皆可dfs.jpg)
- •需要掌握的技巧有:
- · a *, ida *, 蒙特卡洛搜索。





- NOIP2011: Mayan游戏
- 暴搜方案, 然后手动模拟消除、下落。
- 代码量不大。这有何难?
- 对每一层开一组数组或把状态存储为一个类。
- 因为可行解中消除的次数不会太多, 所以复杂度正确。



- NOIP2009: 靶形数独
- 对每行每列每个3*3幻方状态压缩。
- 暴力搜索每个空白位置。通过存储的状态计算出每个位置能填的数字有那些,并保存、更新状态。
- 不需要对权值进行剪枝。只对可填的数字进行剪枝就够了。

