

3.22总结

蓝桥杯历年的题目总是会出现相同考点的，一定要把历年真题每一套都要看一下，考过的知识点一定要好好复习

生死一战 你没有机会懈怠了。2022.3.6

等这个月中旬的时候，把基础课及提高课程的动态规划再刷一遍，动态规划肯定也是今年的主要考点，然后图论的几种最短路算法，最小生成树等基本板子要熟练，赛前20天开始每天起码手要打一遍。另外，多练练暴力搜索，暴力杯还是要把基本功把握好了。

3.5 小白月赛总结与感悟

- 数据范围要注意，最值的设定要看具体情况，比如求最大值，一般情况下直接设置初值为0就可以了，但是有时候往往后面的数据卡的是 $-\text{INF}$ 要注意。
 - 过题的时候要考虑得全面一点，比如这次的括号分割那题，一是括号总体是不是标准的正确的括号，一开始自己想当然直接判断，后面用栈模拟才正确，二是仔细读懂题意，还是括号分割那题，题目原本意思是从中选出来几个进行切割，你以为的是从中选一个割一刀，实际上是可以任选几个出来进行切割，这就要用到快速幂了
 - 动态规划，最重要的是状态表示，状态表示对了，题目基本上就可以过一大半，有时候设置的状态算不出来题目的时候，要突破自己的思维禁锢，想想其它比较好计算的方法，统计数量不重不漏，统计最值时候状态可以重复
-

动态规划是核心 数论 图论基本模型

3.7日总结

基础算法

二分、高精度加减

数学知识

质数、约数、欧拉函数、快速幂、扩展欧几里得算法、求组合数前三种算法

3.8日总结

高精度乘法、除法

前缀和差分

离散化 区间合并

kmp基本

3.9日总结

kmp回顾

单调队列、单调栈

哈希表、字符串前缀哈希

Tire字典树

并查集 维护数量、维护三个相互关系

堆 `priority_queue<int,vector<int>,greater<int>>`

3.11

树形dp都有一个特点就是以某个节点为根节点，算相关情况

二分 看答案是在左边还是右边，如果答案是在区域右边，那么等于号归于l，如果答案在区域的左边，那么等于号归于r

3.13

- 在答案是算一些方案数的时候，要注意最后的答案范围，最好还是开一个long long

3.16

- 当数据到了1e5级别的收，把所有的cin/cout全部替换之后，时间可以快上三倍多

3.17

- 1 欲戴皇冠 必承其重
- 2 博弈个鬼
- 3 好好准备
- 4 踏踏实实的

- 在做状态压缩dp题的时候，关键在于对于状态的一个考虑，从题目当中抽象出来状态的要求，并采取合理的处理/判断状态的方法
- 做状态dp要格外小心和注意，其实任何编程题都应如此，不充分考虑边界甚至连样例都过不了，不考虑完整边界，即使题写出来了，最后一半样例都过不了是很可惜的，尤其是像没开long long这种情况

3.18

- 蓝桥杯除了最后几个可能有点恶心人以外，剩下的题目都是有搞头的
- 在开始写代码之前，先想一下是不是有什么特殊规律，如果是直接对应了某种模型，这个时候也不用着急去直接敲代码，考虑一些求解过程怎么才能相对简单，然后写代码时候，一步步来，最后如果毫无思路，就去想如果暴力该怎么做，暴力之后考虑如何优化，两个大忌，一个是把题目看得太难了，另一个是知道某种模板就直接开始写，也不思考。

3.24 天梯打铁有感

赛后回顾了一下做的题，自己至少有30分的题是没有写出来的

比赛的时候太过于慌张，题目都是看看就丢了，这在ioi赛制是很吃亏的，一些简单的题，也许想一想，算法稍微优化一下都是可以写出来的，自己确和当年考高考一样，慌慌张张，最后一题也没完成，所以说，比赛的时候，写到哪一题，除非非常难，有写出来可能的话还是踏踏实实把当前这道题写完，不要去想后面还有什么题

二就是都怪自己寒假偷了懒，有一道动态维护中位数的是在提高课里面讲过的，几乎就是板子题，要是学了不至于不会写，所以说，未到真正的终点，都不可以懈怠。

三是有一道题映射坐标的时候，自己位运算那块错了，所以一直报错，上次对char数组进行sort也是一样，有时候错误不一定在自己误以为的位置，陷入bug窘境的时候，不要想着什么后果，专注于题目，脑子不要混了

3.25 牛客比赛有感

吸取了上次天梯打铁的经验，这次写题沉稳了很多，基本都是一次性过的，踏实二字，尤为重要！谨记！谨记！

3.28

在数据量达到1e5的时候尽量用scanf输入输出，1e6更需要用了，只要题目数据量稍微卡得严格一点，很可能就会过不了

3.29

- 组合数初始化的时候要从0开始初始化
- [\(3条消息\) 取模运算法则,异或运算法则 天南星南十字星的博客-CSDN博客 取模运算法则](#)
- pow()类型返回的是double类型的
- bfs的时候要注意起点是不是终点的情况
-

模拟题伪代码比较重要

4.8

写好深搜的重要一点就是找到返回的边界，比如某个值达到了多少就要返回，同时如果是枚举某个位置上的数可以填哪些 则要记得恢复现场，即递归dfs之前状态是什么样子的 dfs之后的状态就要恢复成什么样子

