特殊题目处理思路

10circle

2024.8

一些技巧

可以不断的做差分，倘若最后能得到一个常数数列、等差数列这些，就说明原数列和最开始的一些数成组合数关系。
可以调整终端宽度找循环节。
如果循环节有循环节，不断嵌套，可以考虑一些循环节的循环节的循环，找更高一层的规律。
打表验证猜想。大部分猜想都可以验证，相当好用。

例题 P8883

计算 $1 \sim n$ 中能被某个大于 1 的完全平方数整除的数的个数，要求绝对误差不超过 2×10^4。
$n \leq 10^18$，多测。

题解

考虑打出表来，可以发现答案和 n 似乎满足一定的比例关系。尝试输出 $n \times c$，其中 c 是一个打表得到的量。
打表越多即可得到越精确的 c，再上下调整一下就能过了。

例题 P7322

给定常数 k。对于一个长度为 n 的排列 a，定义

$$f(a)=\{\max_{1\le i\le k}\{a_i\},\max_{2\le i\le k+1}\{a_i\},\cdots,\max_{n-k+1\le i\le n}\{a_i\}\}$$

对于一个长度为 n 的序列 a，定义其权值 $w(a)$ 为 a 中不同的数的个数。
现在，ducati 想知道，对于所有长度为 n 的排列 p，它们的 $w(f(p))$ 之和对 998244353 取模的值。

题解

观察到输入两个数，输出一个数，考虑打表。观察到每行第一个最大，而且是 $n \times n!$。考虑横向做差，观察到每行的某一个等于上一行的这个乘 $n+1$，而最后一个则是上一行最后一个乘 n。由此，预处理阶乘后，可以快速计算任意位置的差。随后算完第一个数之后，只需要减去 $(n,1) \sim (n,k-1)$ 的差就好了。

打表所有答案

如题 ()
可以写一些高复杂度的东西，出于优化将跑的慢的部分打表出来。
例题：https://codeforces.com/gym/103469/problem/H
题意是要构造一个图，使得其哈密顿路径数量恰好为 k。
$k \leq 60$，要求图点数 ≤ 20。

分段打表

朴素的打表，指的是在比赛时把所有可能的输入对应的答案都计算出来并保存下来，然后在代码里开个数组把答案放里面，直接输出即可。
注意这个技巧只适用于输入的值域不大（如，输入只有一个数，而且范围很小）的问题，否则可能会导致代码过长、MLE、打表需要的时间过长等问题。
分段打表一般适用于输入一个数，并且可以递推之类的。
这样就可以采用分块的方式隔一段打表一个数，即，整块的答案用预处理的值计算，非整块的答案暴力计算。
例题：P1822

输入输出优化

被卡常而输入输出量大时应优先考虑这个。

找到运行时间瓶颈

可以考虑算每一段的复杂度，如果复杂度接近则应该分开测试时间长短（注释掉其他部分）。
只卡瓶颈的常。其它部分卡了也用处不大，还更难阅读。

合并运算

多试一试，有时优化明显有时没啥用有时负优化。没学过计算机组成，多尝试很重要。

减少不必要的运算

比如少取一次模，少乘一次诸如此类的。

内存不连续访问

内存连续时最快。

循环展开

多试一试，有没用的可能性。

更换数据类型

有时候 unsigned 会快一点，有时换成 int 会快一点。要多试。

递归、归纳子问题法

例题：给定一个只含 L、R 的字符串 s，要求构造一条路径，满足第 i 次转弯是按照 s_i 来的。L 是左转一个直角，R 是右转一个直角。可以自由分配不转弯时走的长度。要求最后回到起点。

upd：找到了此题的最小化面积版本，虽然可能没什么关系：https://loj.ac/p/4744

图上构造考虑 dfs 树

分别考虑树边和返祖边如何构造。

欧拉回路

有时可以利用欧拉回路处理一些奇偶性相关问题。
例题：P8326

题解

这个题看上去就非常欧拉回路的样子，所以考虑建图。
如果就把一个挡板拆两个点的话很不好处理染色，所以考虑点转边，一个挡板是连接两个循环的边然后给边染色。
当然某个挡板的某一面可能不处于某个循环里，将其连向一个特殊点。
有解的条件，首先就是所有循环的长度是 8 的倍数，在这里就是每个点的度数都是 8 的倍数。
直接猜测满足这个条件即有解，尝试构造。
先跑一次欧拉回路，按照顺序奇偶性将边集分成两份，再对这两份边集分别跑一次欧拉回路，就将边集分成四份，分别染四种颜色就好。注意如果边集不连通了就对每个连通块分别跑就好。
容易发现度数一直满足要求。

调整法

先设定一个简单的初始状态，再在其上不断调整使得符合条件。

dp

有时可以 dp 出所有满足构造要求的状态，再沿着可行的状态进行构造。

爆搜/随机化

搜索每一种可能找到可行的构造。
存在有很多很多种解的时候，可以尝试随机化能否随到一组解。
例题：AT_agc012_c

其它建议

有的构造就是很人类智慧，比较运气，没有办法一概而论 ()

根据限制的次数猜测复杂度

和根据数据范围猜复杂度类似，不过有的时候不太靠谱，有的时候这个次数不是一个简单的表达式。

和构造一样，尝试递归至子问题处理

爆搜/dp/随机化得出较优交互策略

例题：NOI2024 D1T2 百万富翁

技巧

观察数据特征，针对每一个点写不同的程序。
可能要跑很久，所以对程序做常数优化很重要。
例题：P6849
尝试拿到更高的分吧！（