

目录

- ① 前言
- ② 技巧
- ③ 搜索的优化
- ④ 考试策略
- ⑤ 真题选讲

前言

搜索和枚举是非常基础的算法，不过在任何级别的比赛都很适用。

但是不同级别的比赛有更复杂，不一样的难度有不一样的技巧。也有很多题是先做一遍搜索没然后进行一些处理。

记忆化

很多搜索的时候重复的状态可以不用处理。

一个简单的方法是用 `map`，如果状态复杂可以用 `map` 套 `vector`。

但有时候速度不够快，怎么办？

`map` 的一个优化：查询的时候用 `find` 不直接用 `[]`。

有没有更快的？

哈希表

哈希表是一个在 OI 内的大部分题目里支持 $O(1)$ 代替 map 的结构。

unordered_map 的内部实现也是哈希表，不过其常数非常大。
经实验，哈希表的模数不应该过大。

记忆化

有些时候记忆化就是一道题的正解，因为实际上是一个 DP，DP 时记录下状态。

有时候搜索的时候必须按照一定的顺序搜索，这样记忆化是对的。

一个经典的例子是插头 DP。

交换顺序

有时候交换顺序可以使得方法可以枚举。
比如是一个区间，可以双指针。
或者很多时候也是常数优化。

最小表示法

试一试

给定一个有 n 个点 m 条边的图，你需要给每个点填一个 $[1, k]$ 的颜色，最终如果有 i 条边连接的两个点连接的颜色不一样，则会产生 c_i 的贡献。

$n \leq 12, m \leq 10^9$ 。

可以考虑先枚举一些量然后计数。
需要知道的有什么？
任意两个点的颜色是否相等？

进行如下枚举。

```
void dfs(int x,int y){  
    if(x==n+1)return;  
    for(int i=1;i<=y;++i)a[i]=y,dfs(x+1,max(y,i+1));  
}
```

这样搜索你就可以知道所有颜色等价类，然后通过组合数和阶乘得出所有等价的方案。

令 c_i 表示 $n = i$ 的时候这样搜索的状态数， $c_{12} = 4213597$ ， $c_5 = 52$ 。

分部分枚举

有很多时候，很多东西是可以分块枚举的，比如图上的搜索，如果有很多连通块，可以分每个连通块单独搜索。

【PKUSC2022 D2T3】雀圣

给定一副麻将牌的初始状态，求最少改变最少的牌使得可以胡牌。

肯定是搜索，不过怎么搜呢？

每种花色之间是独立的，可以搜出来增加 i 个减去 j 个之后能不能使得这个花色合法。

最后背包合并一下。

部分枚举

[NOIP2014 普及组] 子矩阵

定义一个矩阵的价值为所有相邻元素价值差的绝对值之和。
给定一个 $n \times m$ 的矩形，找到一个 $r \times c$ 的子矩阵使得价值最小。

$n, m \leq 16$ 。

对于很多二维问题，且至少有一维很小的时候，可以考虑枚举其中一维的情况，把问题变成一维的问题，或者简化版的二位问题。

这里枚举取了哪些行即可。

部分枚举

给定一个有 n 个点 m 条边的无向连通图，你需要给每个点填一个 $[1, k]$ 的颜色，使得对于任何一条边，其两端颜色不同。

$n \leq 10^5, m \leq n + 5, k \leq 10^5$

首先先求一棵生成树，然后容斥，枚举剩下的边的情况。
然后可以直接最小表示法之后搜索。

折半搜索

AGC026C

给定一个长度为 $2n$ 的字符串。

你需要将所有位置染成红蓝两种颜色。

求有多少种方案，使得红色位置构成的字符串与蓝色部分构成的字符串的反串相同。 $n \leq 18$ 。

考虑分别搜前一半的情况和后一半的情况。
合并的时候是怎么样子的？

UOJ750 小火车

给定 n 个非负整数 $a_1 \sim a_n$ ，以及一个数 p ，保证 $p < 2^n$ 。找到 n 个数 b_i ，满足 $-1 \leq b_i \leq 1$ ， b_i 不全为 0。
 $n \leq 40$ 。

先将题目转化为找到两个和相同的子集。

注意到 $p < 2^n$ ，所以这样的两个子集一定是存在的。

考虑用分治找这个子集，用抽屉原理可以判定是否有解，需要维护的就是和在 $[l, r]$ 区间之内的子集个数，用折半搜索是很好维护的。

特殊复杂度的搜索

随便想的题

给定一个 01 串，假设当前长度为 n ，每次有 $p_{n,i}$ 的概率删掉第 i 个位置，若当前有 c 个 1 则得到 w_n^c 的权值。

求期望得到的权值。

$n \leq 30$ 。

权值概率那些都是乱编的。
考虑直接对于每一个这样的 01 串记忆化。
本质不同的 01 串有多少个呢？

另外还有一些分析方法，比如不超过某个值的状态有多少个，超过某个值的状态有多少个。

一个常用的方式是对于无向图搜索时，每次删掉度数最大的点，然后分连通块，具体复杂度和具体题目有关。

1 前言

2 技巧

3 搜索的优化

4 考试策略

5 真题选讲

剪枝

可行性剪枝。

最优性剪枝。

改变枚举顺序（满足贪心性质的顺序）。

随机化

包括爬山，退火，调整，迭代等。

一般来说，CCF 比赛的大样例和数据是同一个强度，如果能过大样例一般就没问题。

不过考试是一般都是完全正确才满分，所以用正常考试去弄一个随机化是不明智的，除非最后只剩下一道题。

代码长度优化

很多时候写搜索是为了写一个暴力，但一般来说如果一场比赛打很多暴力的话时间不是足够的。

要善用 stl，比如 vector，map，vector<int> 是有重载运算符的，默认的比较方法是字典序比较。

常数优化

能避免递归尽量避免。

调整数组的维度，使得访问更加连续。

在与 01 串有关的题目中，可以考虑压位。

在计数题中，可以考虑用 `unsigned long long` 来优化取模。

搜索和枚举在基本所有考试都很有用，所以这里再谈一下关于写暴力的一些策略，这里分享的大部分是自己的一些经验，如果大家有想分享的也可以一起分享。

什么时候开始写搜索

我个人喜欢如果短时间内没有想到正解做法，并且暴力枚举非常好写的题，可以先写暴力，这样做的好处有很多，比如最后不用补暴力，另外一个好处是可以更方便的找性质，找规律。
另外无论怎么样都必须留足时间写完所有暴力，正式系列的比赛暴力分还是很多的。

写什么样的搜索

写搜索的时候，首先要考虑复杂度能不能通过，除非时间足够不要写理论上过不了的暴力。

如果有跑不满的暴力，可以考虑写。

然后要考虑写的方便，不要浪费太多时间。

不要花太多时间优化不知道能不能卡过的暴力，更重要的是找性质。

比如 NOIP2021 的 T3，找到差分的性质无论怎么搜索都有很多分。

写完搜索之后怎么做

一定要检查是否判断无解。
如果有数据分治记得观察范围。
记得查看特判。
如果写了正解一定要记得对拍。

5 真题选讲

之后的部分题目将会讲解其暴力的写法而不是其正解。

[NOIP2018 提高组] 旅行

给定一棵基环树，你要在上面行走，当走到一个没有走到过的点时记录下这个编号。

最小化最终的字典序。

$n \leq 5000$ 。

经典的基环树问题，一般的基环树问题都可以通过对环边的一些分类讨论进行操作。

首先基环树最终有一条边不需要经过。

此时可以枚举哪一条边不经过，然后在树上可以进行简单贪心。

[CSP-S2020] 儒略日

略

省选 2022 D1T1

略

在考场遇到这种题不要慌，一定要仔细阅读题面中的每一个细节。

写的时候尽量多用函数，减少重复写的时候产生的错误。

如果代码里有分类讨论部分建议自己测试每一处细节。

写完代码之后建议对着题目中的每一个细节检查是否注意到了。

NOIP2009 靶形数独

这种纯搜索题现在已经很少了。
如果遇到的话记得要多剪枝。
有时候可以从密集的地方向稀疏的方向搜，或者稀疏向密集。

NOI2021 量子通信

给定 n 个长度为 256 的 01 串，保证这些 01 串是随机生成的。
给定常数 k 。

q 次询问一个长度为 256 的 01 串，求有多少个串和这个串至多有 k 个位置不同。

$n \leq 4 \times 10^5, q \leq 1.2 \times 10^5, k \leq 15$

首先要知道没有数据随机是做不了的。
然后考虑 15 和 256 之间的关系，是否可以通过某些方法减少枚
举量？

NOI2022 挑战 NPC

给定两棵有根树，节点数量分别为 n_1, n_2 。
求能否从第二棵树上删去一些点，使得两棵树同构。
保证不卡树哈希。

$$n_1, n_2 \leq 10^5, n_2 \leq n_1 + 5$$

首先假如没有 $n_2 \leq n_1 + 5$ 的限制，这个问题是经典的二分图匹配优化 DP 问题，但复杂度过高。

考虑搜索，直接判断 x, y 两个子树是否能够通过删点变得相同。有没有什么贪心的策略呢？

复杂度是什么呢？

Thanks!