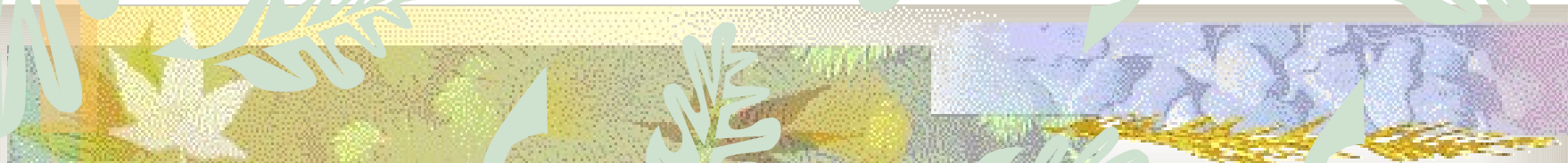


浅谈特殊穷举思想的应用



河北唐山一中 鬲融

穷举的思想

穷举思想是信息学中最重要思想之一，计算机的高速度使其具备了进行穷举的条件。然而，随着图论、数论、动态规划等方法的发展，以及搜索算法的不断改进，穷举似乎越来越不受重视，成为了‘低效’的代名词。

穷举 ~~=~~ 低效？

让我们先来了解一下穷举。

穷举的思想

穷
举

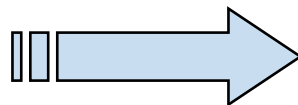
完全穷举

准确理解题意

确定使用穷举思想

明确穷举对象

部分穷举



参变量法

下面先来看一下完全穷举的例子

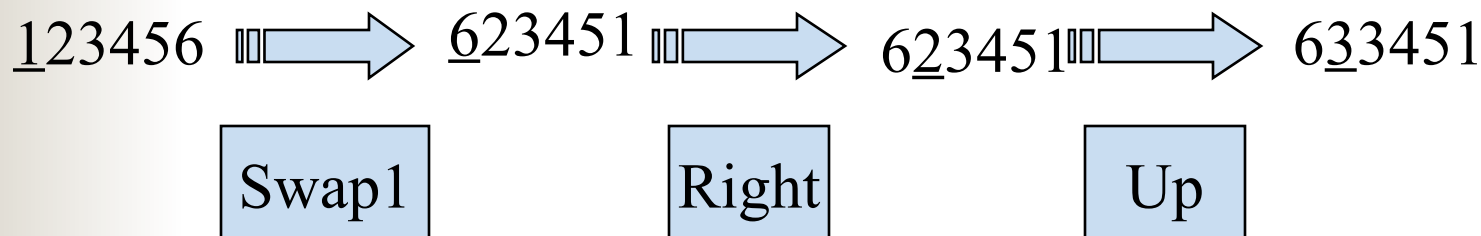
例一 聪明的打字员

题目描述 (NOI2001)

使用一个只有加减 1 (Up/Down)，左右移动光标 (Left/Right)，与 1，6 交换

(Swap0/Swap1) 六个键的键盘，用最少的步数把一个 6 位数转化成另外一个。

例如，初始状态是 123456，要求的状态是 633451，那么最简单的转化方法是：



思路 1 搜索

思路很简单：通过广度优先搜索确定按键顺序和最小按键次数并输出

节点数： **6,000,000** \hat{o} 过大

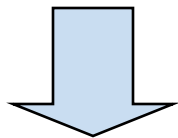
解决： HASH+A* 或双向广度优先

□ 缺点： 实现复杂度太高，而且效率也不高

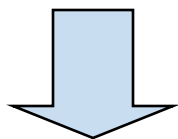
思路 2 使用穷举思想

抓住问题的难点：**Swap0/Swap1**！

要是没有这两个键……直接处理就可以



把这两个键先处理，不影响结果！



穷举这两个键的使用，只有 $6! = 720$ 种情况

思路 2 使用穷举思想

这样我们通过完全穷举得到了一个算法：

把按键过程分为两步

➤通过 Swap0/1 得到一个排列

➤计算这个排列后剩余的步数

一个问题：

并不是所有的位置都可以改变

解决方法：

再次穷举哪些位置可以改变！

时间复杂度： $O(6! * 10) = O(1)$

优秀的算法！



例二 逻辑岛

题目描述：

在逻辑岛上有 3 种居民：永远说真话的神，永远说假话的恶魔和在白天说真话，在夜里说假话的人。

一个社会学家访问了这个岛，但他无法通过外表区分这 3 种居民，所以他只能靠分析居民们说的话来判断他们的种类。

岛上只有五个居民 A， B， C， D， E。

他们说的话只有 3 种

1. I am [not] divine/evil/human/lying.
2. X is [not] divine/evil/human/lying.
3. It is day/night.

居民们说话的总数不超过 50。你的任务就是通过居民们说的话来判断他们的种类以及现在是白天还是黑夜

让我们用人脑分析

一个简单的例子：

A: B is human.

B: A is evil.

A: B is evil.

A 的话有矛盾  B 是神！  A 是恶魔

□ 计算机怎样完成这种推理？

失败！

应用穷举思想

题目特点： 人数少

于是得到了完全穷举的算法：

情况数： $3^5 * 2 = 486$

对每句话进行判断

时间复杂度： $O(3^5 * 2 * s) = O(s)$



小结

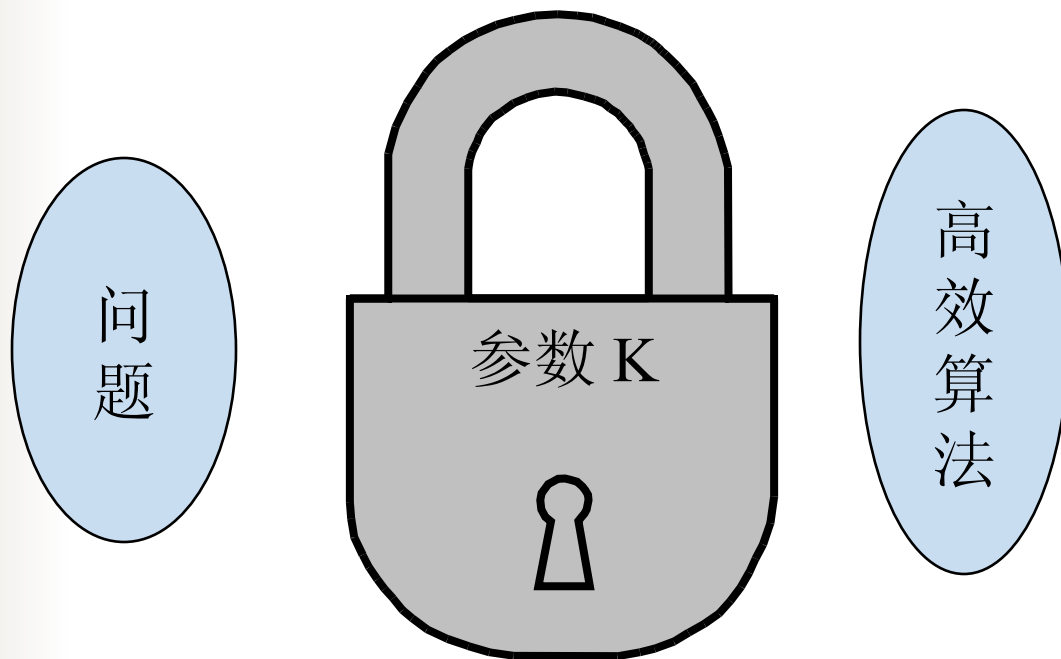
先来比较一下例一的两种算法

算法	时空复杂度	评价
搜索	高, 指数级	未充分利用题目条件
穷举思想	低, 常数级	巧妙选择了穷举对象

穷举或许具有很大的盲目性
，但我们自身是灵活的！

部分穷举思想

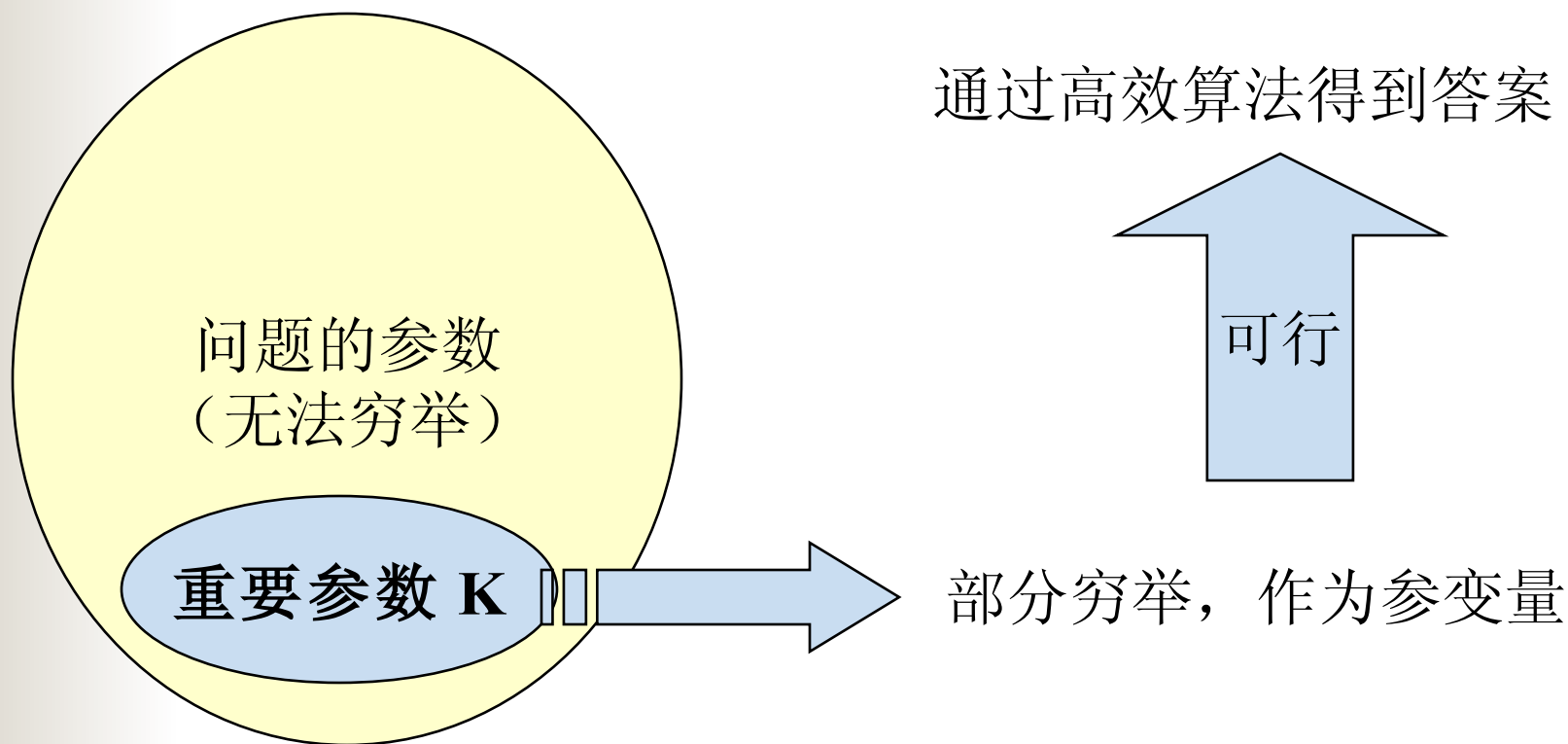
有时候，问题离高效算法只有一步之遥。



我们不知道参数 K，所以我们不能解决问题。

部分穷举思想

使用部分穷举思想的一般步骤。



部分穷举思想特别适用于最大最小问题

最大最小问题

定义：

这类问题中定义一种权值，要求产生一种划分（或其他类似的结构），使划分的每一部分权值的最大值达到最小。

这里最大是权的最大值，最小是划分产生的“最大”的最小。

一个重要的问题

已知一个最大权值，如何快速判断是否有满足要求的划

分？如果我们能找到这个问题的答案，那么我们就可以把最大权值作为参变量，通过对它进行部分穷举和判断，得到结论。

一种常用优化

如果参变量的范围很大，我们就需要利用题目的条件进行优化。

题目最重要的性质是单调性。

单调性是指如果参变量为 x_1 时可行，那么参变量大（小）于 x_1 时也可行，而且题目要求的是参变量的最小（大）值。这时，我们有两种选择：

- 仍然用线形的穷举方法，但是当得到一个可行的解时，就停止
- 改用二分形式的穷举方法。

当然，二分往往可以使时间复杂度减小。

例三 草莓

这是 NOI2003 的题目，相信大家都已经很熟悉。
这里我们要讨论的是树形数据。

初步想法：

既然是求最值，容易想到树的动态规划。可是，
我们很难列出有效的状态转移方程，而且该题数据较大，用动态规划效率太低。

进一步考虑：

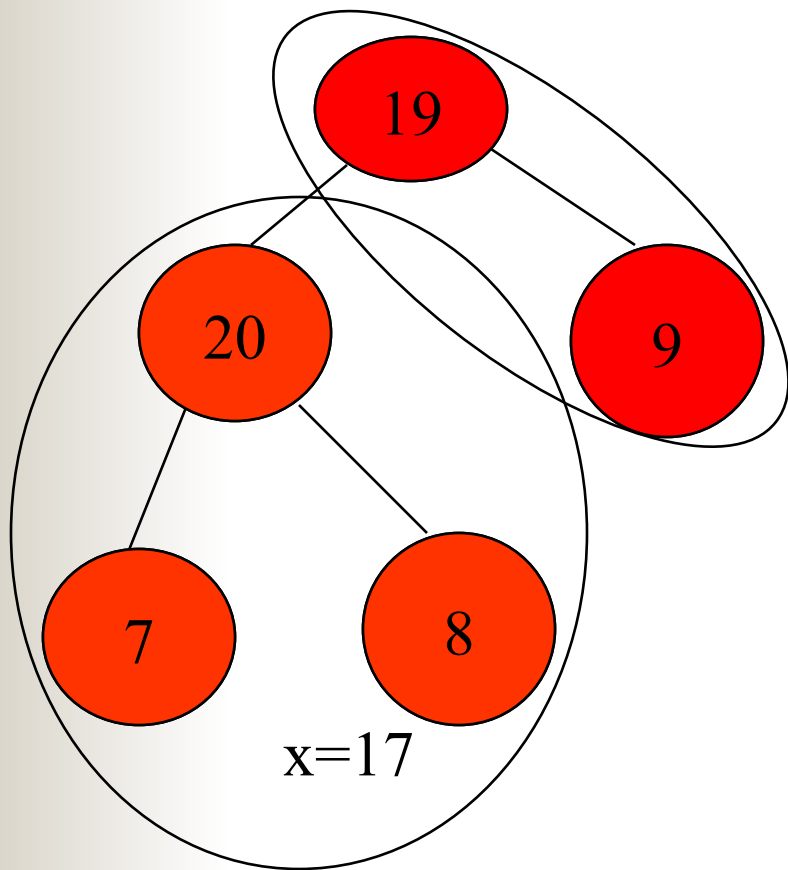
这是一个最小最大问题，能否使用部分穷举？

首先面临的问题：

如果已知一个分割方案所对应的 x ，我们如何去寻找一个解答，或者证明这种分割是不存在的？（问题★）

问题★的解决

这个问题可以用贪心来解决。

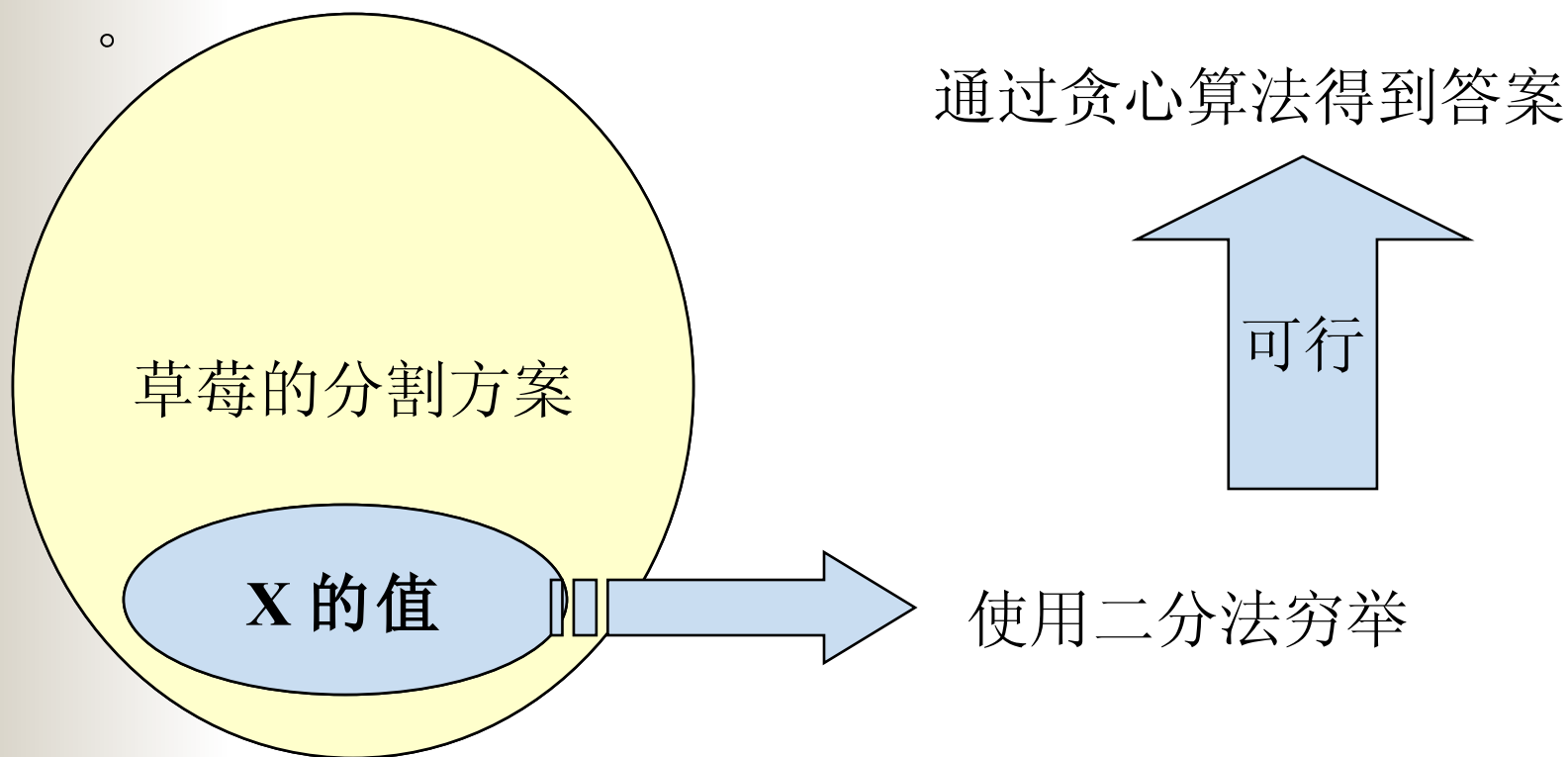


从下向上处理，遇到可以分成一组的就分出来。

对于这个算法的正确性证明请见我的论文。

例三的解决

我们找到了问题★的解法，于是我们就可以通过部分穷举，用二分法穷举 x 的值，然后构造解法。



问题：相应的最大最小问题无法解决。

最大最小匹配

我们来用部分穷举的思想解决一个经典的图论问题。

带权二分图的最大最小匹配问题

已知一个带权的二分图，求一个满足下列条件的匹配：

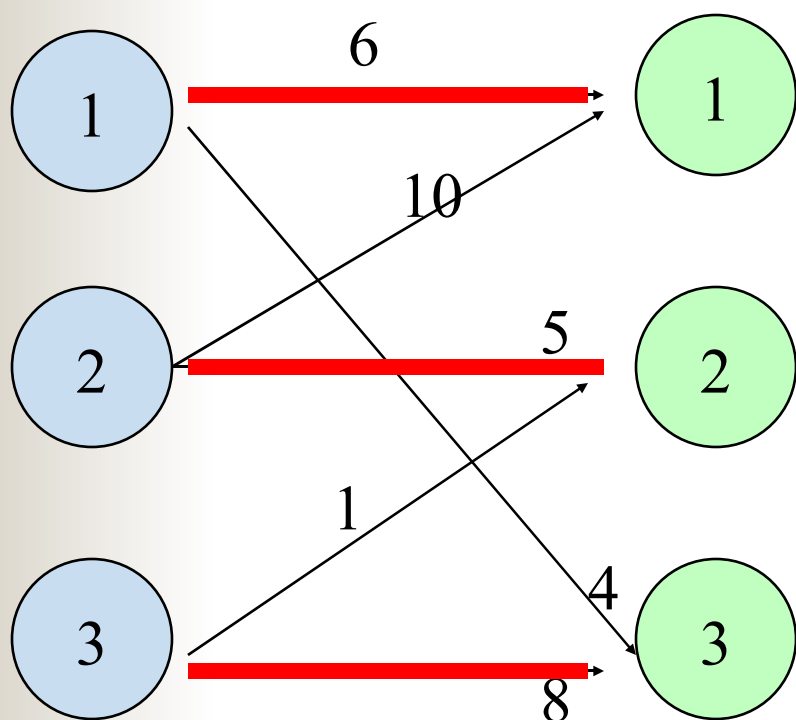
- 这个匹配是一个完备匹配
- 这个匹配是满足第一个条件中，匹配边的最大权最小的匹配。

最大最小匹配的实际意义

这种匹配的计算在日常生活中经常遇到。例如一个公司有 n 个货仓， n 个销售点。已知从第 i 个货仓运送产品到第 j 个销售点所需时间为 $T(i,j)$ ，每个货仓的储藏量和每个销售点的需求量恰好相等。如何在最短的时间内将货物运送到所有销售点？

最大最小匹配

再看一个最大最小匹配问题的实例：

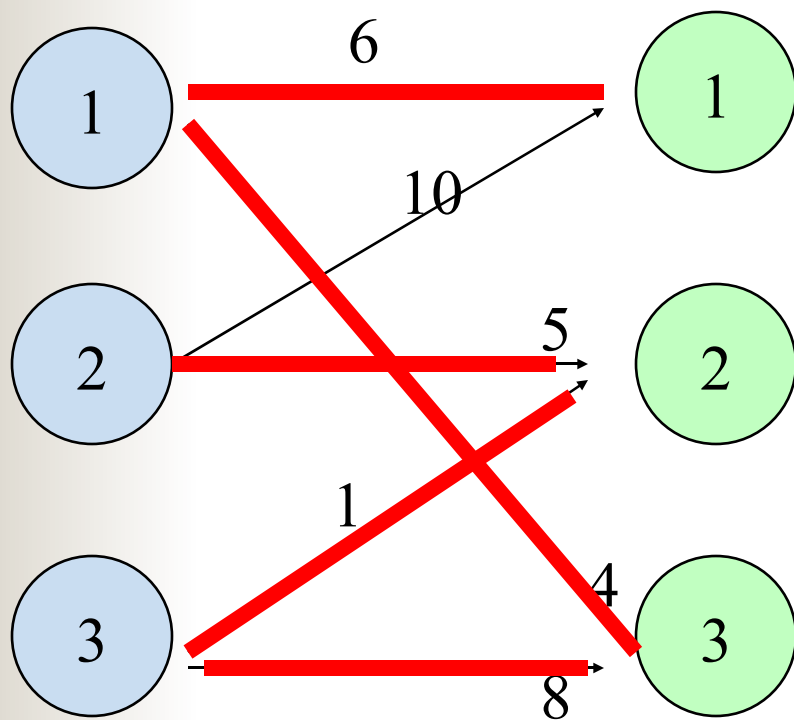


为了清晰，这里只有 6 条边，其他的边权为无穷大，不考虑。

我们可以看到，这个图的最小权匹配不是最大最小匹配。

最大最小匹配：解决

应用部分穷举的思想，我们需要回答这个问题：
如果已知权的最大值，如何产生一个满足题目要求的匹配？



权的最大值： 7

找不到完备匹配！

权的最大值： 8

找到完备匹配！

这就是最大最小匹配。

最大最小匹配：解决

这样我们得到了一个最大最小匹配问题的高效算法：

2. 对最大权进行部分穷举
3. 执行最大匹配的算法寻找完备匹配

时间复杂度： $O(n^5)$ 用线形穷举，但是效果也不错

$O(n^3 \log n)$ 二分穷举。

对这个问题我们还可以进行一些扩展：

- 最小最大匹配……一样的算法
- 带权图的最大最小匹配
- 权最大的带权二分图最大最小匹配


总结

通过分析，我们可以看出：

穷举 \neq 低效

一些结论：

- 对于完全穷举，要正确而巧妙地选择穷举对象，减小穷举量。
- 对于部分穷举，要选择适当的参变量以及后续算法
- 完全穷举和部分穷举是我们解题时的有效思想，我们应对其给予充分的重视，根据题目的具体情况选择应用，不应简单地认为穷举方法效率低而弃之不用。



谢谢大家

谢谢大家

欢迎提问