

# 搜索浅谈

by 冯青同



# 例1 困难的串

## 题意翻译

将一个包含两个相邻的重复子串的子串，称为“容易的串”，其他为“困难的串”。输入正整数 $n$ 和 $l$ ,输出由前 $l$ 个字符组成的，字典序第 $k$ 小的困难的串。

# 例1 困难的串

- 简单的搜索题。
- 一个结论：对于任意合法的串，它的子串也一定是合法的串。
- 直接暴力枚举，考虑在每一次 dfs 时进行 check，看看当前串有没有相邻的重复子串。
- check 时其实没有必要枚举所有的相邻子串。
- 设相邻的两个子串，第一个叫 S1，第二个叫 S2，由上述结论可知，S2 的结尾一定与全串的结尾是相同的。
- 每次搜到一个合法的串就让计数器  $\text{cnt}++$ ，当  $\text{cnt}=\text{n}$  时返回即可。

# 例2 倒水问题 Fill

## 题目描述

有三个容量分别为 $a, b, c$ 升的容器（ $a, b, c$ 都是正整数，且都不超过200），刚开始的时候第一个和第二个杯子都是空的，只有第三个杯子装满了 $c$ 升水。允许从一个容器把水倒入另一个容器中，直到一个容器空了或者是另一个容器满了，允许无限次的进行这样的倒水操作。

你的任务是编写一个程序来计算出最少需要倒多少升水才能让其中某一个杯子中的水有 $d$ 升（ $d$ 是不超过200的正整数）？如果无法做到恰好是 $d$ 升，就让某一个杯子里的水是 $d'$ 升，其中 $d' < d$ 并且尽量接近 $d$ 。如果能够找到这样的 $d'$ ，你还是需要计算出其中某一个杯子达到 $d'$ 升时，最少需要倒多少升水。

## 输入输出格式

### 输入格式：

输入的第一行是一个整数 $T$ ，表示测试数据组数。接下来 $T$ 行，每行4个用空格隔开的整数分别表示 $a, b, c, d$ 。

### 输出格式：

对于每组测试数据，输出一行，包含两个整数，第一个整数表示最少的倒水总量，第二个整数表示目标倒水量（ $d$ 或者 $d'$ ）。



## 例2 倒水问题 Fill

- 这个题有搜索的 tag，但其实用最短路做也是完全 OK 的。
- 讲这个题，主要是想讲讲怎么设计搜索的状态。注意到这个题中的水量很小，所以我们可以把 ABC 三个杯子中的水量作为状态记录下来。这样的话，状态数会是  $200^3$ 。
- 但是考虑一下，其实没有必要记这么多状态，因为我们的总水量确定是  $c$ ，所以我们只需要记 AB 两个杯子中的水量就可以了。这样的话，状态数被压缩到了  $4w$ 。然后这道题其实变成了一道最短路。
- 题解里面都讲使用 BFS，但要把队列换成优先队列。这个东西大家想一想，他不就是我们说的堆优化 Dij 算法吗？
- 由此，我们发现，Dij 实质上其实就是一种变式的 BFS。
- 再思考：那 SPFA 呢？它和 BFS 有什么关系？

## 例3 天平难题

### 题意翻译

给出房间宽度 $r$ 和 $s$ 个吊坠的重量 $w_i$ 。设计一个尽量宽（不超过 $r$ ）的天平，挂着所有挂坠。天平由一些长度为1的木棍组成。木棍的每一端要么挂一个挂坠，要么挂另外一个木棍。设 $n$ 和 $m$ 分别是两端挂的总重量， $a$ 和 $b$ 为两端长度，要让天平平衡，必须满足 $na=mb$ 。

## 例3 天平难题

- 这道题还比较 easy，难点在于如何枚举二叉树。
- 考虑这样的方法：
- 以  $s=6$  为例，最开始只有 6 个节点，分别代表 6 个挂坠，然后考虑节点的合并。这个东西有点类似于那个哈夫曼编码的经典贪心，就是每次取出来两个节点  $a$  和  $b$ ，建一个新的节点  $x$ ，把  $a$  和  $b$  都设成  $x$  的儿子。容易证明，这样的操作最多进行 5 次。
- 建出来树以后，自顶向下 DFS 求解答案即可。



## 例3 天平难题

- 刘汝佳老师还给出了一种状压枚举子集来枚举二叉树的做法。具体来说，自定义一个类型 `node`，记录 `l` 和 `r` 分别表示向左和向右伸展的最长距离，然后开一个 `vector` 来记录子集 `s` 的所有可能的树的 `l` 和 `r`，合并的时候枚举子集即可。时间复杂度来分析一下？每棵可行的二叉树都只会被遍历  $2n$  次，那可行的二叉树有多少种？这里我们用 `dp` 来算一下呗？
- 设 `dpi` 表示一个大小为 `i` 的集合按照每次划分为两个集合的方式，最终划分为 `i` 个大小为 1 的集合的方案数，那么  $dp_i = \sum C(i, k) * dp_k * dp_{i-k}$ ，这个东西能算吗？
- $dp_1 = 1$
- $dp_2 = 2$
- $dp_3 = 12$
- $dp_4 = 48 + 24 + 48 = 120$



# 例4 旋转游戏

## 题目描述

如图 1 所示，有一个“#”形的棋盘，上面有 1, 2, 3 三种数字各 8 个。给定 8 种操作，分别为图中的 A ~ H。这些操作会按照图中字母与箭头所指明的方向，把一条长度为 8 的序列循环移动 1 个单位。例如下图最左边的“#”形棋盘执行操作 A 时，会变为图中间的“#”形棋盘，再执行操作 C 后会变为图中最右边的“#”形棋盘。

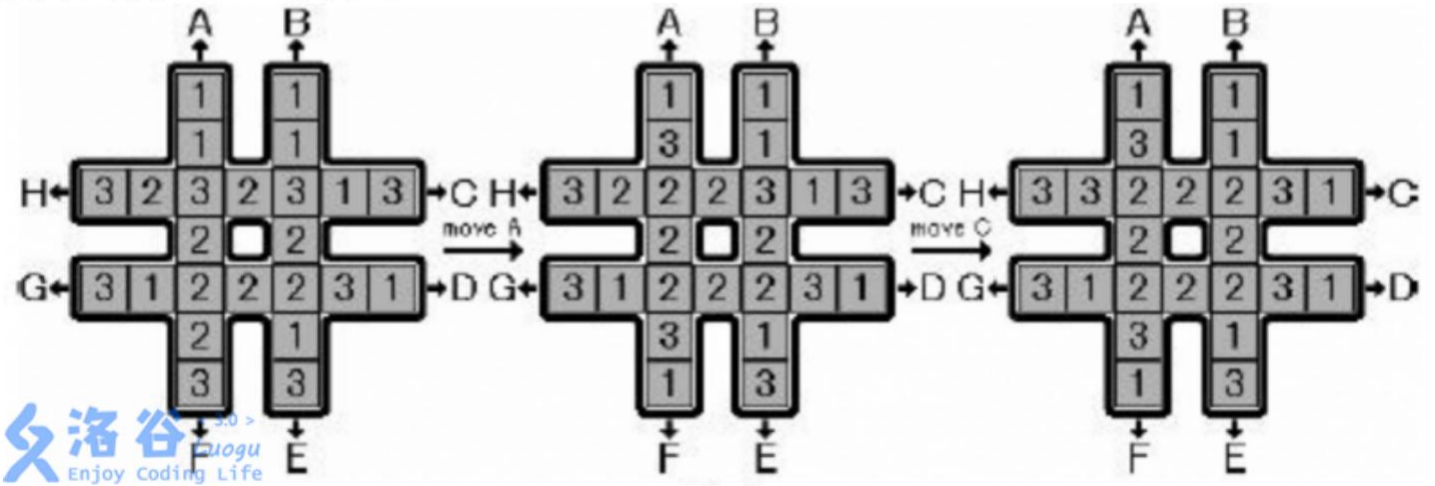


图 1

现给定一个初始状态，请使用最少的操作次数，使“#”形棋盘最中间的 8 个格子里的数字相同。

## 例4 旋转游戏

- 难点有二：
- 1、如何存图、操作；
- 2、搜索没有一个明显的上界，不能使用 DFS；状态很难记录，不能用 BFS。
- 对于第一个问题，把图像输入给出的那样编号，转化为一维数组。对于每一种操作，因为实际上就是一次轮换变化，所以把操作涉及的格子都拿出来，按照顺序排列好，然后就可以在一维数组上直接进行操作了。

## 例4 旋转游戏

- 考虑第二个问题，这种既不能用dfs又不能使用bfs的搜索题，我们会使用迭代加深搜索。
- 什么是迭代加深搜索呢？它是基于这样一个性质来考虑，就是dfs的解答树每增加一层，它的新的一层的节点的量级会以非常快的速度增加，以至于前面所有层的和也不如它大。
- 所以我们可以为dfs的搜索人为的设计一个上限  $d$ ，我们枚举  $d$  从一到正无穷，搜索层数达到  $d$  时，停止搜索。假如我们在上限为  $d$  的时候能搜出答案，那么答案就是 $d$ 了；如果不能，就让  $d++$ ，然后接着搜。
- 对于这个题，就是我们先设一下，让操作次数最多为 $d$ ，如果你当前的次数已经多于 $d$ ，我们就不继续搜索了。



## 例4 旋转游戏

- 这样的话，时间复杂度还稍有不够，所以我们要考虑采用启发式搜索的方法来优化。
- 什么是启发式搜索？简单的说就是设计一个估价函数，让我们可以剪掉更多的枝。
- 考虑估价函数  $h$  能够算出当前状态到达目标状态，至少还需要多少次操作，如果当前的操作数  $+h > d$ ，也停止搜索。
- 注意这样一个问题：设当前状态达到目标状态实际上需要  $g$  次操作，我们估计的  $h$ 。必须比  $g$  小而且越接近  $g$ ，剪枝效果更好。
- 这里直接令  $h = 8 - \text{中间 } 8 \text{ 格里出现次数最多的数字的出现次数}$  即可。

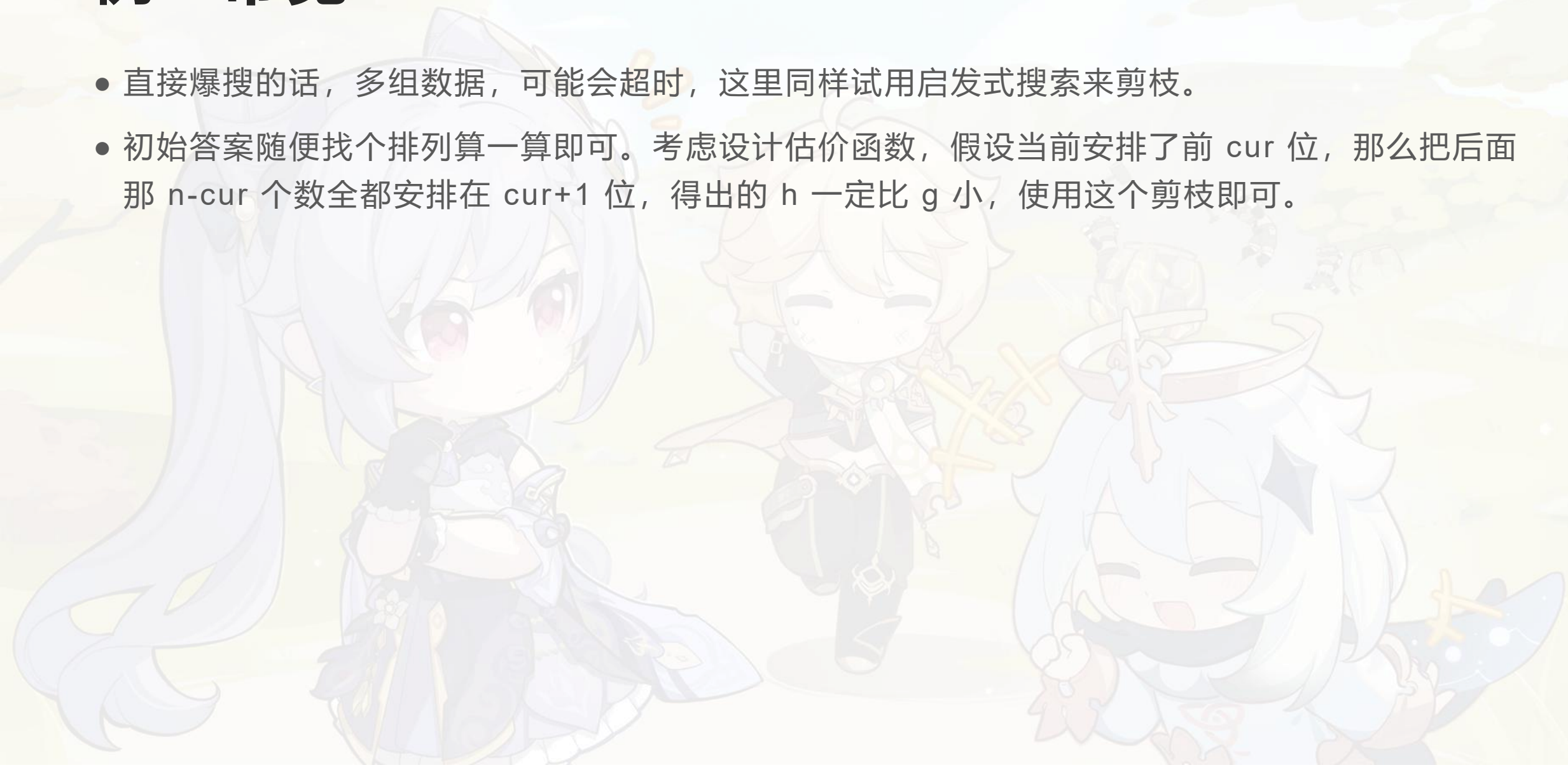
## 例5 带宽

### 题意翻译

题意： 给一个最多8个结点的无向图，把结点重排后对于图中每条边 $(u,v)$ ， $u$ 和 $v$ 在排列中的最大距离称为该排列的带宽。求带宽最小的排列。

## 例5 带宽

- 直接爆搜的话，多组数据，可能会超时，这里同样试用启发式搜索来剪枝。
- 初始答案随便找个排列算一算即可。考虑设计估价函数，假设当前安排了前  $cur$  位，那么把后面那  $n-cur$  个数全都安排在  $cur+1$  位，得出的  $h$  一定比  $g$  小，使用这个剪枝即可。



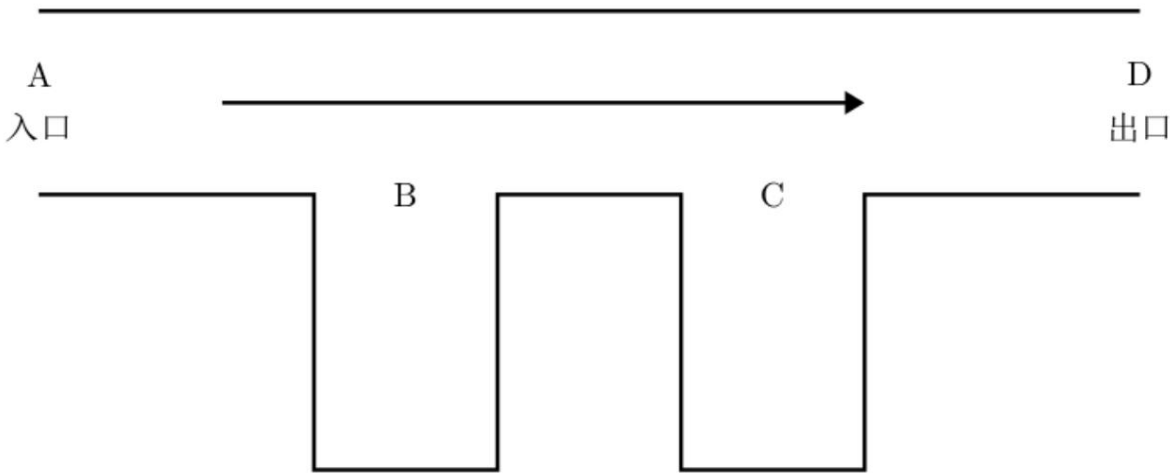


# 例6 单向双轨道

## 题目描述

[M+](#) [复制Markdown](#) [🔍](#) [展开](#)

如图所示，某火车站有 B、C 两个调度站，左边入口 A 处有  $n$  辆火车等待进站(从左到右以  $a, b, c, d$  编号)，右边是出口 D，规定在这一段，火车从 A 进入经过 B、C 只能从左向右单向开，并且 B、C 调度站不限定所能停放的车数。



洛谷

从文件输入  $n$  及  $n$  个小写字母的一个排列，该排列表示火车在出口 D 处形成的从左到右的火车编号序列。输出为一系列操作过程，每一行形如  $h, L, R$  的字母序列，其中  $h$  为火车编号， $L$  为  $h$  车原先所在位置（位置都以 A, B, C, D 表示）， $R$  为新位置。或者输出 NO 表示不能完成这样的调度。

## 例6 单向双轨道

- 一共有 6 种操作，直接搜索是可能超时的，这里依然采用启发式迭代加深搜索。
- 深度  $d$  的可能范围是  $n \sim 3n$ ，估价函数  $h$  设置为当前仍然没有进站的火车的总数即可。
- 但在这里，我们要继续考虑一下剪枝优化：
  - 1、假如上一步是  $A \rightarrow C$ ，那么下一步一定不会是  $C \rightarrow D$ （举个例子，上一步的抵达车站和这一步的出发车站相同的都是不可能最优的），因为不如直接一步  $A \rightarrow D$
  - 2、假如要动一辆车  $x$ ，并且  $x$  可以进  $D$ ，那么  $x$  直接进  $D$  是最优的。

# 例7 八城堡问题

## 题目描述

仰慕皇后的“城堡们”，也想学着“八皇后”站出“八城堡”的站位。但是就像城堡需要有地基一样，他们也不能随意布局，只能被建造在高地上，洼地则不行（容易倒）。同时任意两个城堡不能布局在同一行或者同一列上，以防两城堡之间发生部落冲突。

“城堡们”不够聪明，于是邀请聪明的你来帮助他们设计城堡布局的方案，快来帮帮他们吧！

“城堡们”在一个给定边长 $N$ 的正方形棋盘上按一定方式排列，棋盘上用 **H** 字符表示高地：可以建造城堡； **L** 字符表示洼地：不可以建造城堡。

“城堡们”需要被布局在高地上，同时任意两个城堡都不能放在正方形棋盘的同一行或者同一列上（要不然就会打架了）

请你设计一个程序，帮“城堡们”算出来对于给定大小和地形的棋盘，要摆放 $M$ 个城堡的话，一共有多少种方案呢？



# 例7 八城堡问题

- $n=8$
- 随便搜即可



# 例7 八城堡问题

- $n=14$
- 这里直接搜就不太行了，我们介绍一种优化复杂度的方法：双向搜索，或者是折半搜索。
- 双向搜索也是基于 dfs 解答树的节点增长情况而想出的一种优化方法。我们可以把  $n=14$  的问题转化为两个 7 行 14 列的搜索，即上半棋盘和下半棋盘分别搜索出答案，最后考虑把两个答案拼一块儿就行了。
- 咋拼呢？我们采用状态压缩的思想。记  $f_{i,s} = \sum g_t$ ， $t$  是  $s$  的子集且  $t$  的大小为  $i$ 。  $g_t$  表示上半棋盘中，属于  $t$  的列都放了城堡的方案数。

# 例7 八城堡问题

- $n=20$
- 状态压缩记忆化搜索





# 例8 P2324 骑士精神

## 题目描述

[M↓ 复制Markdown](#) [📄 展开](#)

在一个  $5 \times 5$  的棋盘上有 12 个白色的骑士和 12 个黑色的骑士， 且有一个空位。在任何时候一个骑士都能按照骑士的走法（它可以走到和它横坐标相差为 1，纵坐标相差为 2 或者横坐标相差为 2，纵坐标相差为 1 的格子）移动到空位上。

给定一个初始的棋盘，怎样才能经过移动变成如下目标棋盘：



## 例8 P2324 骑士精神

- 启发式迭代加深搜索。
- 乐观估价函数设计为不在右上角的黑马与不在左下角的白马的总数之和。
- 直接搜索即可。



# 例9 P1763 埃及分数

## 题目描述

[M+](#) 复制Markdown [🔍](#) 展开

来源：BIO 1997 Round 1 Question 3

在古埃及，人们使用单位分数的和（形如  $\frac{1}{a}$  的， $a$  是自然数）表示一切有理数。如： $\frac{2}{3} = \frac{1}{2} + \frac{1}{6}$ ，但不允许  $\frac{2}{3} = \frac{1}{3} + \frac{1}{3}$ ，因为加数中有相同的。对于一个分数  $\frac{a}{b}$ ，表示方法有很多种，但是哪种最好呢？首先，加数少的比加数多的好，其次，加数个数相同的，最小的分数越大越好。如：

$$\begin{aligned}\frac{19}{45} &= \frac{1}{3} + \frac{1}{12} + \frac{1}{180} \\ \frac{19}{45} &= \frac{1}{3} + \frac{1}{15} + \frac{1}{45} \\ \frac{19}{45} &= \frac{1}{3} + \frac{1}{18} + \frac{1}{30} \\ \frac{19}{45} &= \frac{1}{4} + \frac{1}{6} + \frac{1}{180} \\ \frac{19}{45} &= \frac{1}{5} + \frac{1}{6} + \frac{1}{18}\end{aligned}$$

最好的是最后一种，因为  $\frac{1}{18}$  比  $\frac{1}{180}$ ,  $\frac{1}{45}$ ,  $\frac{1}{30}$  都大。

注意，可能有多组最优解。如：

$$\begin{aligned}\frac{59}{211} &= \frac{1}{4} + \frac{1}{36} + \frac{1}{633} + \frac{1}{3798} \\ \frac{59}{211} &= \frac{1}{6} + \frac{1}{9} + \frac{1}{633} + \frac{1}{3798}\end{aligned}$$

由于方法一与方法二中，最小的分数相同，因此二者均是最优解。

给出  $a, b$ ，编程计算最好的表达方式。保证最优解满足：最小的分数  $\geq \frac{1}{10^7}$ 。



## 例9 P1763 埃及分数

- 迭代加深搜索，枚举最多使用  $d$  个分数拼成  $a/b$ 。
- 剪枝方法：搜索时按照分母由小到大搜。如果 当前的分数\*剩余的数的个数 比还要拼的部分少，那么就直接返回不可能即可。



## 例10 P1514 引水入城

- 第一问直接把第一行的所有点都放入队列，然后跑 BFS 判断连通性即可。
- 第二问，有这样的结论：第一行的任意一个点所能覆盖的最后一行的点一定是连续的一段。
- 然后直接做一个线段覆盖的贪心问题即可。

