谈谈搜索题

杭州学军中学　周子鑫

2017 年 3 月 21 日

ZJOI2017 交流群

133135071

Section 1

这节课要讲点什么呢

这节课要讲搜索。

那什么是搜索呢？

搜索算法是利用计算机的高性能来有目的的穷举一个问题解空间的部分或所有的可能情况，从而求出问题的解的一种方法。现阶段一般有枚举算法、深度优先搜索、广度优先搜索、A* 算法、回溯算法、蒙特卡洛树搜索、散列函数等算法。在大规模实验环境中，通常通过在搜索前，根据条件降低搜索规模；根据问题的约束条件进行剪枝；利用搜索过程中的中间解，避免重复计算这几种方法进行优化。

通俗来说，大家平时写的暴力，都在广义的" 搜索" 概念内。

今天我们探讨的搜索题既有具有严格分析的时间复杂度的科学搜索题，也有无法计算准确时间复杂度的玄学搜索题。

Section 2

搜索热身

来看看两道简单的热身题。

例题 1 (NOIP2012 普及组 文化之旅)

有一位使者要游历各国，他每到一个国家，都能学到一种文化，但他不愿意学习任何一种文化超过一次（即如果他学习了某种文化，则他就不能到达其他有这种文化的国家）。不同的国家可能有相同的文化。不同文化的国家对其他文化的看法不同，有些文化会排斥外来文化（即如果他学习了某种文化，则他不能到达排斥这种文化的其他国家）。

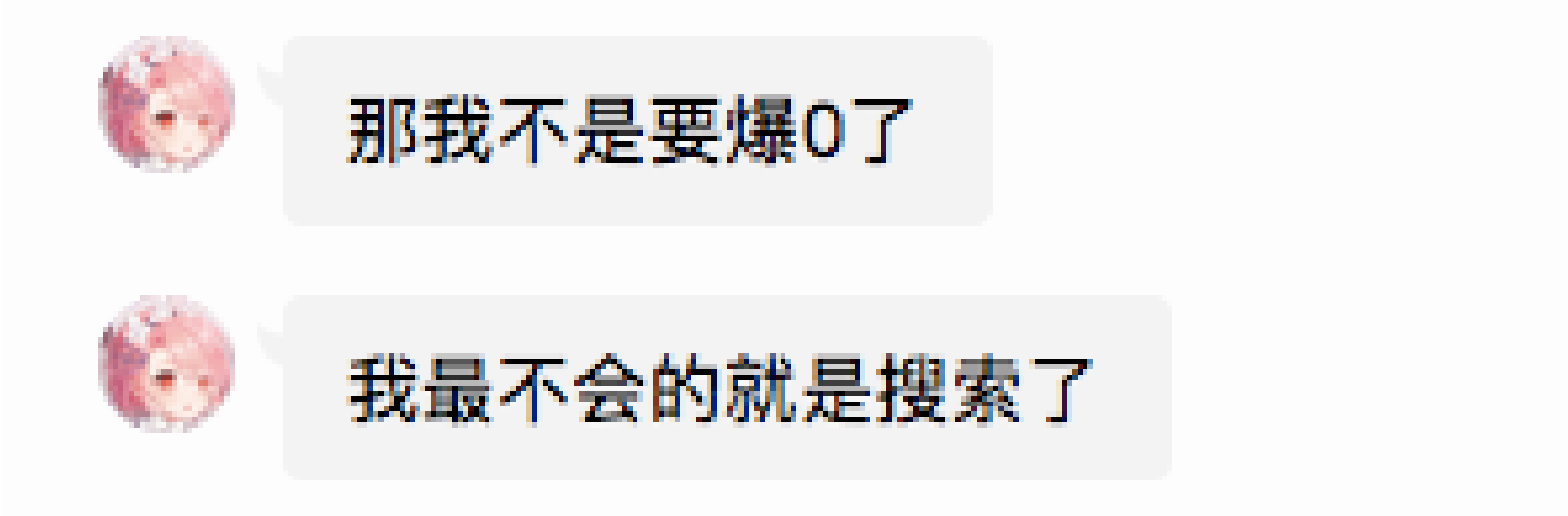
现给定各个国家间的地理关系，各个国家的文化，每种文化对其他文化的看法，以及这位使者游历的起点和终点（在起点和终点也会学习当地的文化），国家间的道路距离，试求从起点到终点最少需走多少路。

国家数，文化数 ≤ 100

这题一眼看上去就是 NPC 的。

然而它真的是 NPC 的吗？

是的，它确实是 NPC 的。



现在我们请这位大佬上台讲讲他对这个题的看法。

如果按照 CCF 的官方数据，有 100 种错误的方法可以过掉此题。

比如，只考虑相邻两个城市之间的文化是否排斥，或者只是找出到从起点到终点第一条合法的道路。

当时我在考场上写了个 A*，因为一个地方写错了，只拿了 30 分。

例题 2 (NOIP2015 斗地主)

牛牛最近迷上了一种叫斗地主的扑克游戏。斗地主是一种使用黑桃、红心、梅花、方片的 A 到 K 加上大小王的共 54 张牌来进行的扑克牌游戏。每一局游戏中，一副手牌由 n 张牌组成。游戏者每次可以根据规定的牌型进行出牌，首先打光自己的手牌一方取得游戏的胜利。

现在，牛牛只想知道，对于自己的若干组手牌，分别最少需要多少次出牌可以将它们打光。请你帮他解决这个问题。

$n \leq 23$

有没有哪位大佬愿意上台来表演一下。

Hint: 最后再考虑单牌和对子。

接下来讲几个搜索中的通用优化。

Section 3

折半搜索

如果问题能分成两个相对独立的部分，且两部分搜索结果能很方便合并，那么就可以利用折半搜索来大大提高搜索的效率。

来看道题形象地理解一下。

例题 3 (方程的解数)

已知一个 n 元高次方程

$$k_1 x_1^{p_1} + k_2 x_2^{p_2} + ... + k_n x_n^{p_n} = 0$$

假设未知数均为不大于 M 的正整数，求这个方程解的个数。

$n \leq 6, M \leq 150$

将前半部分的搜索结果存入 hash 表，再去搜后半部分。

例题 4 (ZJOI2005 九数码问题)

这是一个很古老的游戏了：有一个 3×3 的活动拼盘，方格上写有 0 8 这九个数字。利用拼盘背后的旋钮，游戏者每次可以进行以下两种操作之一：

1. 将拼盘外围的 8 个方格按顺时针挪一个位置。
2. 将中间一行向右移动一个位置，最右边的方格被移到最左边。

给你一个拼盘的初始状态，你能用最少的操作次数把拼盘变成目标状态吗？

例题 5 (Rikka with Sequence II)

众所周知，萌萌哒六花不擅长数学，所以勇太给了她一些数学问题做练习，其中有一道是这样的：勇太有一个长度为 n 的数列，现在六花可以从中选出若干个数（不可以不取）。她的方案需要满足她选出的所有数的平均数小于等于中位数。现在勇太想让六花求出满足条件的方案数。

注：数列长度为偶数的时候，中位数为排序后最中间的两个数的平均数。

当然，这个问题对于萌萌哒六花来说实在是太难了，你可以帮帮她吗？

$n \leq 40, A_i \leq 10^9$

先把所有数从小到大排序，接着可以枚举中位数是哪两个数的平均数，假设是 l 和 r（长度为奇数的情况相当于 $l = r$），然后我们给数列中所有数减去平均数，那么这时的方案数就等于从 $[1, l)$，$(r, n]$ 这两个区间中选出相同数目的数使得它们的和小于 0。

我们可以把待选的数给单独拿出来（即把区间 $[l, r]$ 的数删掉），如果原来它的下标小于 l，把么权重为 1 否则权重为 -1。问题就相当于给出若干个数，你要选出一些数使得它们的权重和为 0 且权值和小于 0，这是一个经典的 meet in middle 的问题。我们可以把数均等的分成两部分，分别处理出两部分中的所有选取方法，然后根据权重分组，每组排序之后就能统计答案了。

上述算法的时间复杂度是:

$$\sum_{i=1}^{n}(n-i+1)\times 2^{(n-i)/2}\times(n-i)/2=O(2^{n/2}\times n^2)$$

然而这样做可能还是会 TLE 的，各位大佬有没有什么更好的想法。

我们发现我们没有必要排序，我们可以在枚举的时候使用归并的方法，直接让每组有序，这样就能减少一个 n 的复杂度。

我们发现，折半搜索的技巧在非玄学的搜索题中都有很多的应用，只要是答案能方便合并的题目基本上都可以采用这个技巧来让搜索的层数变成原来的一半，大幅优化算法。

Section 4

启发式搜索

启发式搜索又称为有信息搜索，它是利用问题拥有的启发信息来引导搜索，达到减少搜索范围、降低问题复杂度的目的，这种利用启发信息的搜索过程称为启发式搜索。

启发式搜索中应用最为广泛的搜索技巧当属 A* 和 IDA* 了，前者是 BFS 的启法式版本，后者是前者的迭代加深版本。

A* 和 IDA* 算法中，对每一个状态 x 引入了一个估价函数 $f(x) = g(x) + h(x)$, 其中 $g(x)$ 是目前状态的实际代价，而 $h(x)$ 是目前状态到目标状态的估计代价。

在 A* 算法中，估价函数的作用是调整搜索顺序让最优的解最先搜索到。而在 IDA* 中，估价函数作为最优性剪枝出现。

为了保证搜索结果的正确性，$h(x)$ 不能大于当前状态到目标状态的最优值。

特殊地，当 $h(x) = 0$ 时，A* 和 IDA* 算法退化为一般的 BFS 算法与迭代加深算法。

例题 6 (十五数码问题)

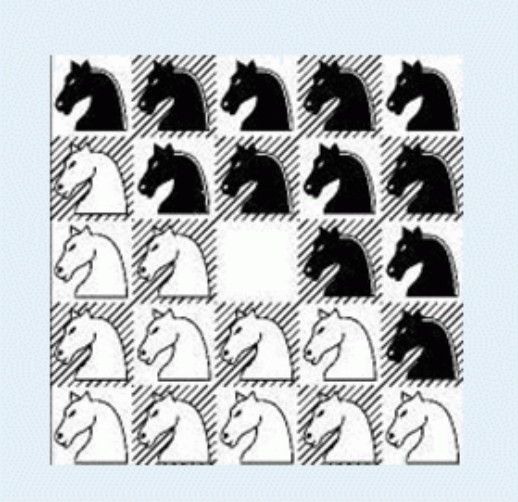
在 4×4 的棋盘，摆有十五个棋子，每个棋子上标有 *1* 至 *15* 的某一数字，不同棋子上标的数字不相同。棋盘上还有一个空格，与空格相邻的棋子可以移到空格中。要求解决的问题是：给出一个初始状态和一个目标状态，找出一种从初始转变成目标状态的移动棋子步数最少的移动步骤。

这题是经典的八数码问题的一个加强版，因为状态数的大大增多，就不能直接裸着 BFS 了，要考虑优化。

首先考虑 A* 算法，但是因为空间问题，只能使用 IDA* 算法。

例题 7 (SCOI2005 骑士精神)

在一个 5×5 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：



为了体现出骑士精神，他们必须以最少的步数完成任务。

从名字就可以看出，启发式搜索是个比较玄学的东西，具体效果和估价函数有直接关系。

如果能判断出题目就是一个玄学搜索或者没什么好办法的时候，不妨可以试一试。

Section 5

Dancing Links

Dancing Links 是由 Knuth 提出，解决一类覆盖问题的通用框架，它本质上只是属于常数优化，但是它优化效果非常好且适用面较广，故我将它单独介绍。

大家在写双向链表的时候，有删除一个结点 x 的操作。

$$L[R[x]] \leftarrow L[x]$$

$$R[L[x]] \leftarrow R[x]$$

现在如果要撤消这个删除操作呢？

$$L[R[x]] \leftarrow x$$

$$R[L[x]] \leftarrow x$$

只要这样就行了，嘿嘿 

可是这踏马有什么卵用呢...

为什么要在把一个结点删除掉以后再把它放进来呢？

一个典型的应用是在 DFS 的时候，还原之前的链表状态。

Dancing Links 被设计出来解决的问题一般如下：

给定一个由 0 和 1 组成的矩阵，问能否找到一个行的集合，使得集合中每一列都恰好包含一个 1。例如下面的矩阵

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

这个矩阵的 1, 4, 5 行就是这样的一个集合。

已知精确覆盖问题是一个 NPC 问题，那该怎么样来搜呢？

算法 1 (Algorithm X)

如果 A 是空的，问题解决；成功终止。

否则，选择一个列 c（确定的）。

选择一个行 r，满足 $A[r,c]=1$ （不确定的）。

把 r 包含进部分解。

对于所有满足 $A[r,j]=1$ 的 j，

　　从矩阵 A 中删除第 j 列；

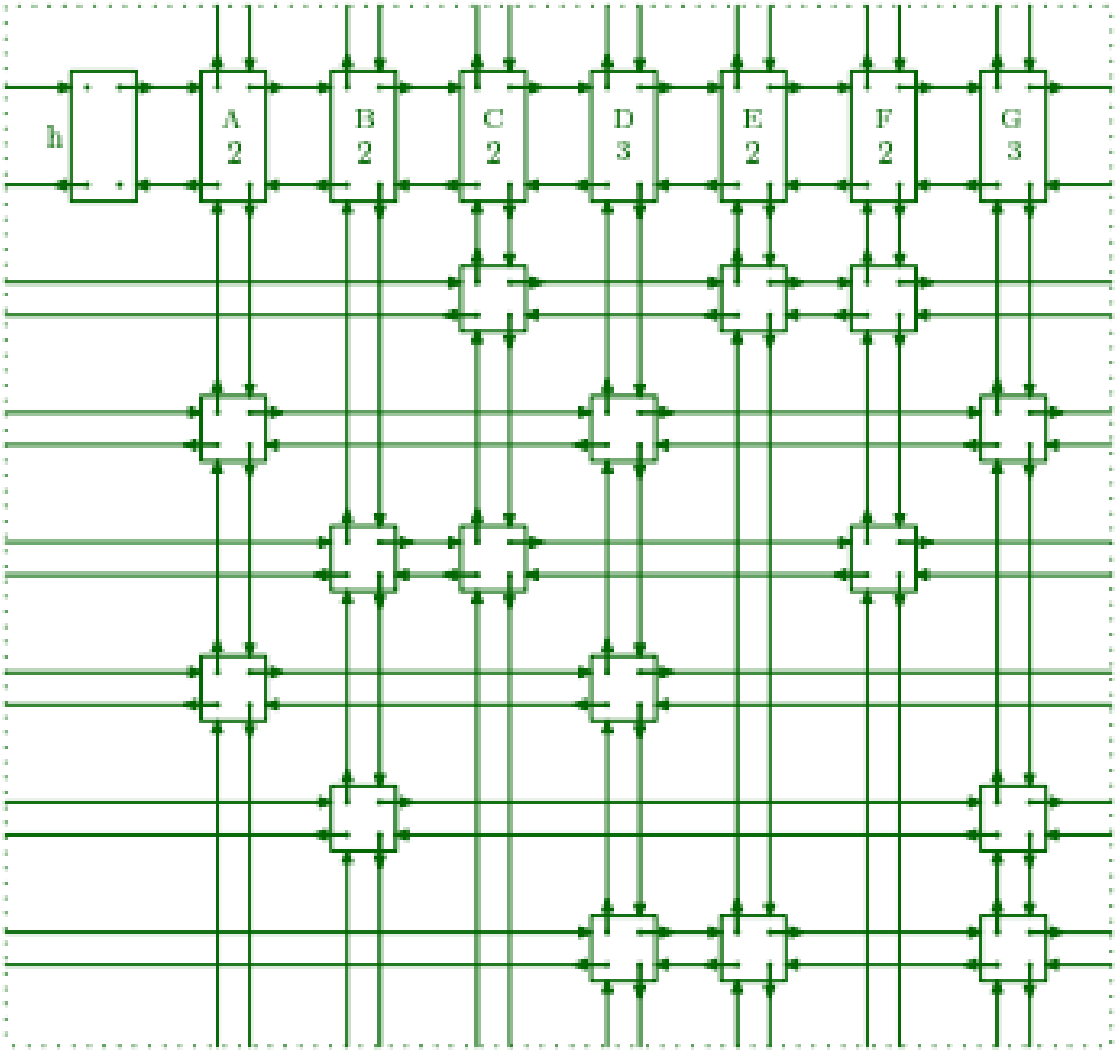
　　对于所有满足 $A[i,j]=1$ 的 i，

　　　　从矩阵 A 中删除第 i 行。

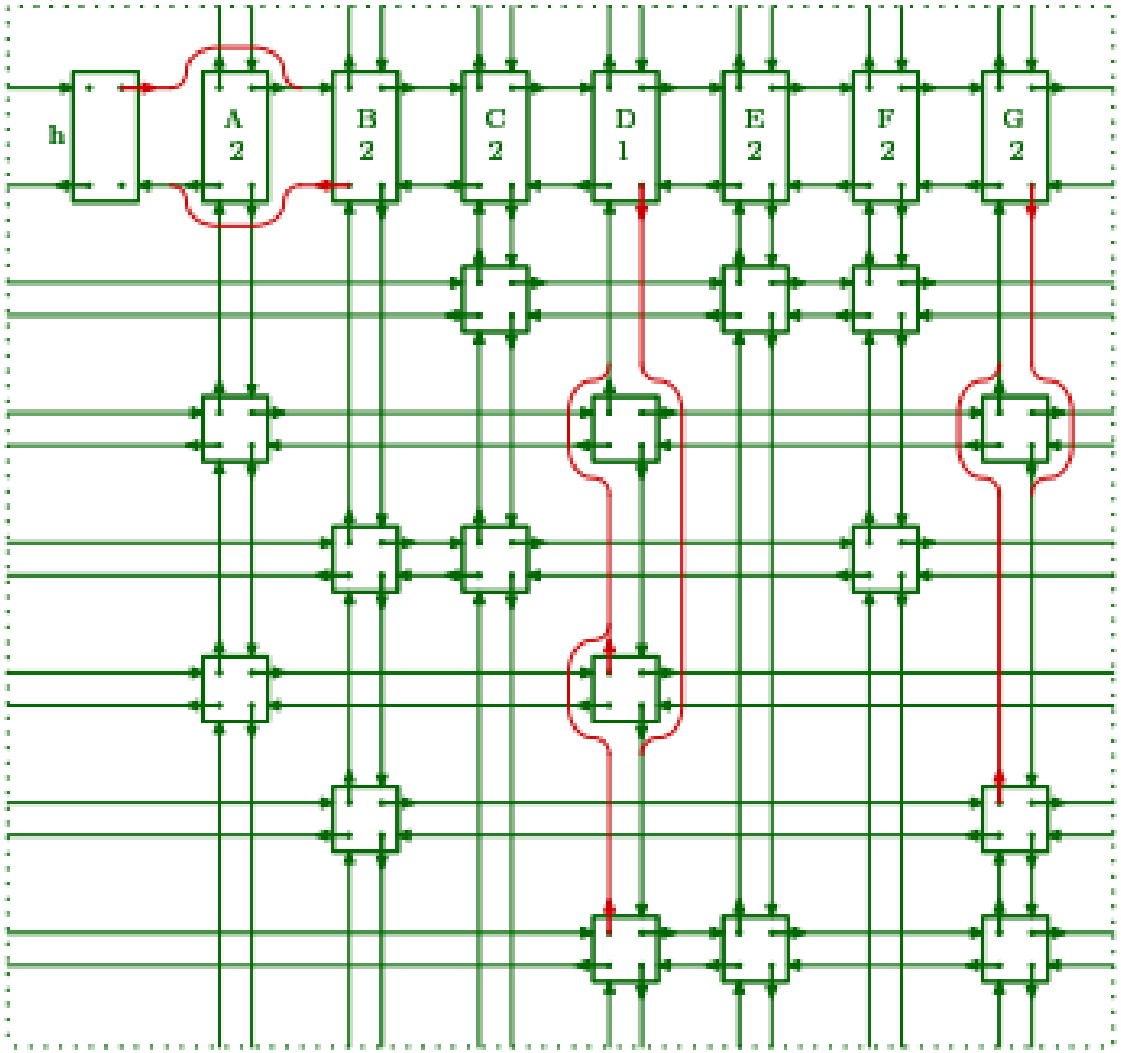
在不断减少的矩阵 A 上递归地重复上述算法。

可以看到这个 X 算法一直在干一个事情就是删除一行或者删除一列。很自然地，我们想到了使用 Dancing Links 来进行删除与还原这两个操作。使用 Dancing Links 来维护的 X 算法被称为 DLX 算法。

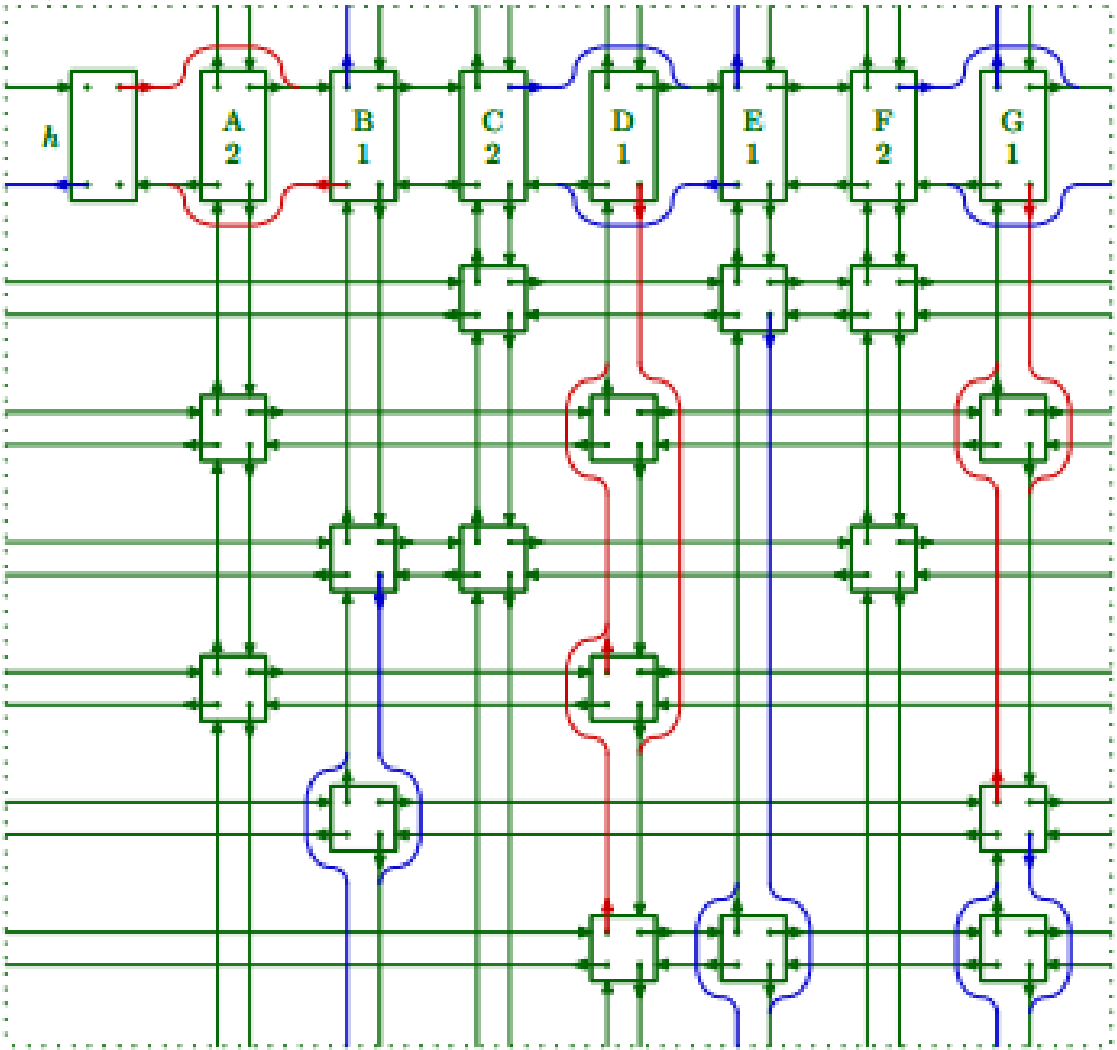
这里我们使用上下两维的双向链表，U, D, L, R 四个指针来表示每个 1 在矩阵中的关系。



删除一列的时候:



删除一行的时候:



加上一个简单的启发优化。

在每层 DLX 开始的时候，都需要选择一列 c，那么这个时候，选当前矩阵该列的 1 最少的那一列。

啊哈，这样你就学会了纯正的 DLX 算法。来看看几个例题吧。

例题 8 (NOIP2009 靶型数独)

靶形数独的方格同普通数独一样，在 *9* 格宽 *9* 格高的大九宫格中有 *9* 个 *3* 格宽 *3* 格高的小九宫格。在这个大九宫格中，有一些数字是已知的，根据这些数字，利用逻辑推理，在其他的空格上填入 *1* 到 *9* 的数字。每个数字在每个小九宫格内不能重复出现，每个数字在每行、每列也不能重复出现。但靶形数独有一点和普通数独不同，即每一个方格都有一个分值，而且如同一个靶子一样，离中心越近则分值越高。

输出可以得到的靶形数独的最高分数。如果这个数独无解，则输出整数*-1*。

搜搜搜。考虑怎么转化成精确覆盖问题。

例题 9 (hdu4979 A simple math problem)

有一种彩票，共有 n 个数字，其中 r 个为获奖数字，每张彩票上选择 m 个不同数字，若 m 个数字中包含 r 个获奖数字则获奖。问至少要买多少彩票才能保证获奖。其中 $n \geq m \geq r$。举例:

$n = 6, m = 3, r = 2$。只用买 6 张彩票即可。*1,2,3*，*1,4,5*，*1,3,6*，*2,4,6*，*2,5,6*，*3,4,5*。

$n \leq 8$

可以暴力把所有的可能的彩票给枚举出来，再枚举所有可能的获奖情况。

发现，这好像不是一个精确覆盖问题，这其实是个多重覆盖问题。

那要怎么解多重覆盖问题呢？

稍微修改下 DLX 的 remove 和 resume 过程即可。

代价是，DLX 的魔力大减，虽比爆搜快点，但已不可和精确覆盖的 DLX 同日而语了。

没办法，TLE 得不要不要的，尝试加继续加入启发式的优化，引入之前讲过的 IDA*。

我们最后得到的算法是，使用修改过的 DLX 再加上 IDA* 的框架进行搜索，如果估价函数不是太不靠谱的话，我们已经得到了一个在不考虑问题特殊性情况下几乎是最优的多重覆盖解法了。

然而还是 TLE... WTF?!

那怎么办办呢..

嘿嘿

打表...

第二个办法是，因为这个题只有一组输入是很难在时限内跑出的，而且这个题的答案可能的范围不大，如果你不在意罚时的话，就可以不停地试这一组输入的答案，直到对了为止。

实际上现场过这题的两个队都是用第二个办法的。

例题 10 (N 皇后问题)

如何能够在 $n \times n$ 的国际象棋棋盘上放置 n 个皇后，使得任何一个皇后都无法直接吃掉其他的皇后？为了达到此目的，任两个皇后都不能处于同一条横行、纵行或斜线上。

$n \leq ???$

N 皇后问题中，行和列的限制都很容易就能转化为精确覆盖问题。但是斜的方向呢，好像不一定每一条对角线时候要有皇后。

这时候应该怎么办呢？

搜的时候不去管那些列就行了咯。

经过上面的学习，我相信大家都已经掌握了使用 DLX 解决精确覆盖、多重覆盖，以及有些列不一定要覆盖的问题。

并且使用 DLX 解决上面这些 NPC 问题，基本是在不考虑问题特殊性情况时，已知在实践中最快的算法了，以后遇到长得很像这类问题的题目，可以努力往覆盖问题上面套。

Section 6

剪枝

剪枝，顾名思义就是指在 DFS 的过程中判断出不可能成为答案的分支并停止向下 DFS。

我把剪枝放到最后的原因是，剪枝是所有搜索优化技巧中最难的，需要对题目性质深入的挖掘，普遍套路也是最少的。但剪枝可以说是搜索题精髓之所在。也能最大程度帮助大家提高“能力”。

剪枝一般有两大类:

1. 可行性剪枝。将不可能产生合法解的分支舍弃掉。一般可配合启发式搜索顺序使用。

2. 最优性剪枝。将不可能产生比当前最优解更优解的分支舍弃掉。一般可配合估价函数使用。

直接来看几道例题吧。

例题 11 (NOI2003 智破连环阵)

给出地图上 N 个目标点，M 台武器（只能用一次）。每个武器启动后，将距它距离不超过 K 范围内的符合要求的目标全部摧毁（符合要求：摧毁目标必须按编号顺序，而且同一时刻只有一个武器是启的），求摧毁全部目标需要动多少武器。

$n, m \leq 100$

Hint:

1.dp 求估价函数，最优性剪枝。

2. 利用匹配信息求新的区间的可行范围，可行性剪枝。

例题 12 (SDOI2015 排序)

给定一个长度为 2^n 的排列，有 n 个操作，第 i 个操作为【将序列分成 2^{n-i+1} 段，每段长 2^{i-1}，然后任选两段交换】，每个操作最多用一次，求有多少操作序列能把序列排好序。

$n \leq 12$

Hint:

1. 一个合法方案，交换任意两个操作不会改变合法性。
2. 如果小段区间不合法，那么大段肯定不行。可行性剪枝。

Section 7

题目选讲

例题 13 (围栏问题 by dzy)

在一片草原上，有 n 只兔子无忧无虑地生活着。这片草原可以划分成 $m \times m$ 的方阵。每个方格内最多有一只兔子。一位饲养员负责喂养这些兔子。为了方便，她需要用篱笆建造最多 k 座围栏，将草原上的兔子全部围起来。围栏需要满足以下条件：

(1) 必须沿着网格线建造；

(2) 每座围栏是一个不与自身重叠或相交的封闭回路；

(3) 各座围栏之间互相不重叠、不相交；

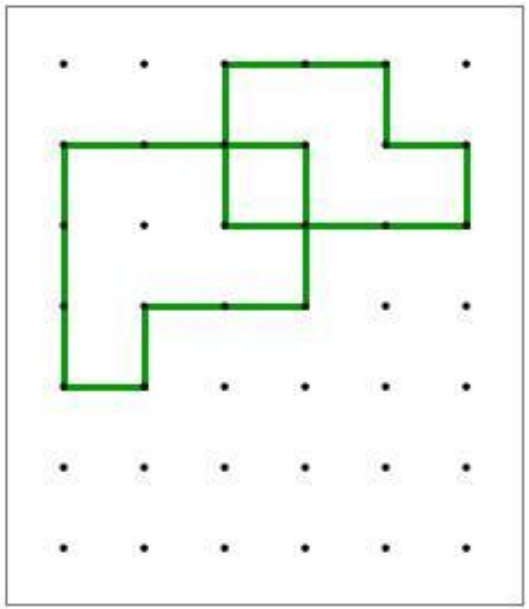
(4) 一座围栏不能被围在另一座围栏里面。

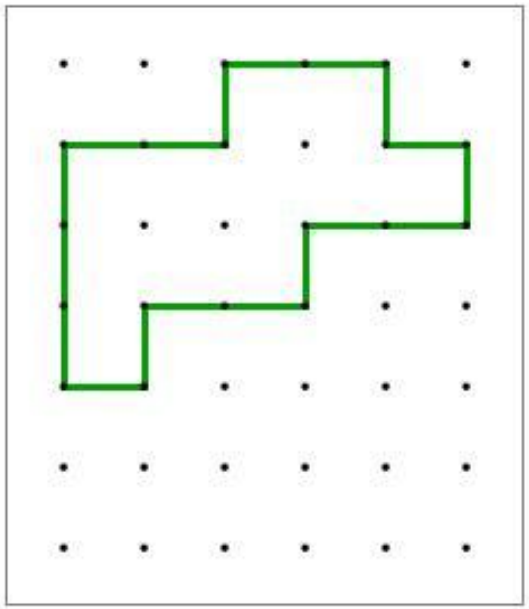
请你帮助饲养员计算一下围栏总长度的最小值。

$k \leq n \leq 16, m \leq 1000$

首先，两个围栏互相包含的情况不可能在最优解中出现（把里面那个拆掉可以节省篱笆，得到一个更优解）。

其次，两个围栏相交的情况也一定不是最优，如下图所示，把重叠的部分拆掉可以得到更优解。





矩形显然不会比不规则形状的更劣，因此只要考虑矩形就行了。

只需要考虑所有的极小矩形，因为矩形之间相交不优，因此问题就转化成了一个精确覆盖问题，DLX 既可。

例题 14 (SRM555 MapGuessing)

已知有一个初始值均未知的长度为 n 的 01 图灵机和一个磁头，定义以下 4 种操作：

1. 将磁头左移一位
2. 将磁头右移一位
3. 将磁头当前所在位置的值赋为 *0*
4. 将磁头当前所在位置的值赋为 *1*

现在给出一个长度为 len 的操作序列和某个状态，求有多少初始图灵机状态，满足存在一个磁头的初始位置，使得磁头在移动中始终不移出图灵机，且存在一个中间状态等于给出的状态。

$n \leq 36, len \leq 555$

由于 n 的范围比较大，不能直接枚举图灵机的状态，考虑从磁头的初始位置入手。

假设枚举了磁头的初始位置，那么我们就可以通过模拟操作，得到磁头是否会移出图灵机以及被修改的位置，如果被修改的位置对不上，那么这种初始位置一定是不合法，否则考虑最后一次所有被修改的位置都能对应给出状态的时刻 (此时被修改的位置一定最多)，所有被修改的位置都可以任意选，而未被修改的位置就要和给出状态一致。

现在已经得出了一个磁头初始位置能够满足的所有初始序列，但是一个初始序列可能会被多个磁头满足，所以需要去重。

设磁头初始位置 i 能自由选择的位置集合为 A_i，所有 A_i 的可重集为 S，根据经典的容斥定理，可以得到正确的答案为 $\sum_{S' \subset S} (-1)^{|S'|+1} 2^{|P|}$，其中 P 为 S' 中所有 A_i 的交集。

可以在 $O(2^n)$ 的复杂度内解决。

如果在搜索时，当 P 为空集时就停止搜索，就可以通过此题了，下面来分析算法的正确性。

考虑一个磁头在操作中的移动范围 $[L, R](L \leq 0, R \geq 0)$，设移动范围长度为 l，则可行的磁头位置数量至多为 $n - l$，而每个磁头 i 可能的被修改位置为 $[L + i, R + i]$，即个数最多为 l.

由于集合要非空，而可能被修改位置是一段区间，那么每次可以选择的那些集合之间最大的左端点距离差不能超过 l，也就是说能够使得交集非空的磁头位置之间最大的距离差不超过 l，那么能够使得交集非空的磁头集合数量不会超过 $O(n * 2^l)$.

由以上分析可得搜索的最大复杂度为 $O(min(2^{n-l}, n * 2^l))$，则时间复杂度约为 $O(\sqrt{n2^n})$.

时间复杂度 $O(\sqrt{n2^n})$，空间复杂度 $O(n^2)$。

感谢冯哲同学本题的题解!

例题 15 (NOI1997 文件匹配)

一种简单的正则表达式由英文字母（区分大小写）。数字及通配符"*"和"?"组成，"?"代表任意一个字符，"*""则可以代表零个或任意多个字符。

写一个程序，寻找一个正则表达式，使其能匹配的待操作字符串最多，但不能匹配任何不进行操作的字符串。注意你所找到的最优正则表达式的长度应当是最短的。如果有多个长度最短的最优正则表达式，则其中任意一个都是允许的。

字符串个数 ≤ 250，字符串长度 ≤ 8

Hint:

1. 记录下每个串当前所有可能匹配到的位

2. 如果所有的正文件全部不能严格匹配，则再搜索下一位已经没有意义。

3. 如果前缀匹配的正文件的个数小于已有答案，再搜索下一位答案一定不会增加。

4. 如果匹配串末字符是 *，当前答案小于最优答案，继续搜索一定是不会使答案增加。

5. 启发式搜索顺序，先搜分支少的。

例题 16 (NOI1998 软件安装盘)

给出一张 n 个点的 *DAG*, 每个点有点权，你需要将这些点分成若干组，并满足：

1. 每个组内点权和不超过 M

2. 存在一个拓扑序使得每个组内的点连续。

在满足上面条件的情况下，让组数尽可能小。

$n \leq 100$

Hint:

对 DAG 拓扑排序，确定搜索顺序

IDA*

例题 17 (NOI1999 生日蛋糕)

7 月 *17* 日是 *Mr.W* 的生日，*ACM-THU* 为此要制作一个体积为 N 的 M 层生日蛋糕，每层都是一个圆柱体。设从下往上数第 i 层蛋糕是半径为 R_i，高度为 H_i 的圆柱。要求 $R_i > R_{i+1}$ 且 $H_i > H_{i+1}$。由于要在蛋糕上抹奶油，为尽可能节约经费，我们希望蛋糕外表面（最下一层的下底面除外）的面积 Q 最小。

请编程对给出的 N 和 M，找出蛋糕的制作方案（适当的 R_i 和 H_i 的值），使 S 最小。

经典搜索题。

1. 根据剩余的层数计算可能的最小体积和最大体积进行可行性剪枝。

2. 根据估价函数进行最优性剪枝。

考虑满足当前剩余体积 V，要求的是所构成的其余层的面积的下界 S'。

只有一层时：

$S = R^2 + 2 * R * H = V * (1/H + 1/R + 1/R)$

$S \geq V * 3 * (1/(H * R^2))^{1/3}$

$S \geq 3 * V^{2/3}$

因为如果有多层，面积显然会损失，因此 $3 * V^{2/3}$ 是剩余体积为 V 时面积的下界。

例题 18 (NOI2000 算符破译)

古梅文明的数学文字一共有 *13* 个符号，与 *0,1,2,3,4,5,6,7,8,9,+,*,=* 这 *13* 个数字和符号（称为现代算符）一一对应。为了便于标记，我们用 *13* 个小写英文字母 $a,b,\cdots m$ 代替这些符号（称为古梅算符）。但是，还没有人知道这些古梅算符和现代算符之间的具体对应关系。

在一个石壁上，考古学家发现了一组用古梅算符表示的等式，根据推断，每行有且仅有一个等号，等号左右两边为运算表达式（只含有数字和符号），并且等号两边的计算结果相等。

假设这组等式是成立的，请编程序破译古梅算符和现代算符之间的对应关系。

等式个数 ≤ 1000, $5 \leq$ 等式长度 ≤ 11

先枚举 $+$, *, $=$ 三个符号对应的字母。

然后一行一行去搜，如果遇到没出现过的字母，就枚举它的值。

除开普通的剪枝外，最重要的一个剪枝是：$=+*$ 这三个确定后一块块数字的位数也就确定了。对于每一行，我们算出左右两边的可能的答案位数范围，然后判断有无交集，没有就不合法。

例题 19 (ZJOI2014 璀璨光华)

金先生有一个女朋友——没名字。她勤劳勇敢、智慧善良。金先生很喜欢她。为此，金先生用 a^3 块独特的水晶制作了一个边长为 a 的水晶立方体。他要将这个水晶立方体送给他见过最单纯善良的她。没名字收到了礼物后果然不一会儿就根据说明将水晶立方体拼好了。

没名字发现，有 n 块水晶在漆黑安静的夜晚会随机向上下左右前后六个方向的一个发出光。被光照到的水晶显得格外好看。没名字给每一块不会发光的水晶定义了一个好看程度。水晶立方体在夜晚中的好看程度就是每块被光照到的水晶的好看程度之和。没名字想知道，水晶立方体在夜晚中的好看程度的最小值和最大值。

$n \leq 8, a \leq 70$

例题 20 (ZJOI2014 2048)

提交答案题，写个 *2048* 的 *AI* 告诉你随机数生成方式.

祝大家省选顺利!