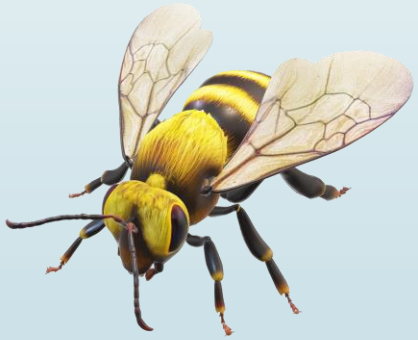
搜索专题

Shinetism

YALI

July 20, 2019



搜索？

- 搜索是曾经风靡一时的算法
- 零几年的NOIP几乎必考搜索
- 搜索的特点：
  - 暴力专供（几乎所有题目的第一档部分分都是搜索）
  - 便于综合其他算法（搜索常常是正解的一部分）
  - 时间复杂度玄学（各种乱搞剪枝）
- 因为最后一点，这些年NOIP很少考复杂的搜索
- 比如2012年普及组——

P1078 文化之旅

题目背景

本题是错题，后来被证明没有靠谱的多项式复杂度的做法。测试数据非常的水，各种玄学做法都可以通过（比如反着扫），不代表算法正确。因此本题题目和数据仅供参考。

于是我们只需要了解复杂度正确的搜索~

剪枝是个技术活. 和贪心算法一样, 剪枝没有固定规律, 需要大胆猜想和不断尝试. 所以今天不单独讲剪枝.

记忆化搜索

复杂的状态转移？

记忆化搜索

- 有人说，记忆化搜索就是DP，为什么还要单独拿出来讲？
- 因为记忆化搜索便于实现更加复杂的状态转移！
- 比如高维的、带状态压缩的DP用记忆化搜索实现起来就更方便.
- 这里举一个题

T1 SCOI 2008 着色方案

- Description:
  - 有 n 个木块排成一行, 从左到右依次编号为 $1 \sim n$. 你有 k 种颜色 的油漆, 其中第 i 种颜色的油漆足够涂 c_i 个木块. 所有油漆刚好足够涂 满所有木块, 即 $c_1 + c_2 + \cdots + c_k = n$. 相邻两个木块涂相同色显得很 难看, 所以你希望统计任意两个相邻木块颜色不同的着色方案.
- Restraints:
  - $n \leq 15, k \leq 5$

T1 SCOI 2008 着色方案

- 设 $search(a,b,c,d,e,l)$ 表示当前有第一种油漆能涂 a 次，第二种油漆能涂 b 次…前一个颜色为 l 的方案数即可.

迭代加深搜索

深度+广度！

迭代加深搜索

- 迭代加深搜索，就是在 dfs 时限制搜索深度不超过 lim. 依次加大 lim 进行多次深度优先搜索.
- 这样，搜索时会估计其他枝叶，不会有 dfs 的缺点.
- 同时，也不会有广度优先搜索难编写、空间占用大的缺点.
- 虽然重复搜索，但是鉴于每增加一次 lim，则搜索的时间复杂度会以指数级别增加，所以重复搜索的时间可以忽略.

T2 棋盘染色

- Description:
  - 有一个 5×5 的棋盘，上面有一些格子被染成了黑色，其他的格子都是白色，你的任务的对棋盘一些格子进行染色，使得所有的黑色格子能连成一块，并且你染色的格子数目要最少. 读入一个初始棋盘的状态，输出最少需要对多少个格子进行染色，才能使得所有的黑色格子都连成一块（四联通）.

T2 棋盘染色

- 我们限制搜索深度, 也就是枚举更改颜色的格子数为 lim.
- 依次增大 lim, 接着搜索更改 lim 块能否联通.
- 设 $search(x, y, k)$ 表示当前到达 (x, y) , 已经填了 k 块.

A*搜索

智能化!

A*搜索

- A*搜索是一种启发式搜索.
- 什么是启发式搜索？
  - 没有一定的搜索次序（不是深搜也不是广搜）
  - 根据具体问题选择较优的搜索次序
- 一个经典的例题是带障碍网格的最短路.
- 用曼哈顿距离作为启发函数，优先搜索离终点直线距离更近的点.

T3 一道经典题

- Description:
  - 给出一个 n 个点、m 条边的图 G，求两点之间的第 k 短路.
- Restraints:
  - $2 \leq n \leq 5000,\ 1 \leq m \leq 200000$

T3 一道经典题

- 先逆向跑一边Dijkstra, 得到每个点离终点距离, 作为启发函数.
- 然后正向跑 A*, 当终点第 k 次出队时, 得到的就是 k 短路.
- Time Complexity: $O(nk \log n)$

IDA*搜索

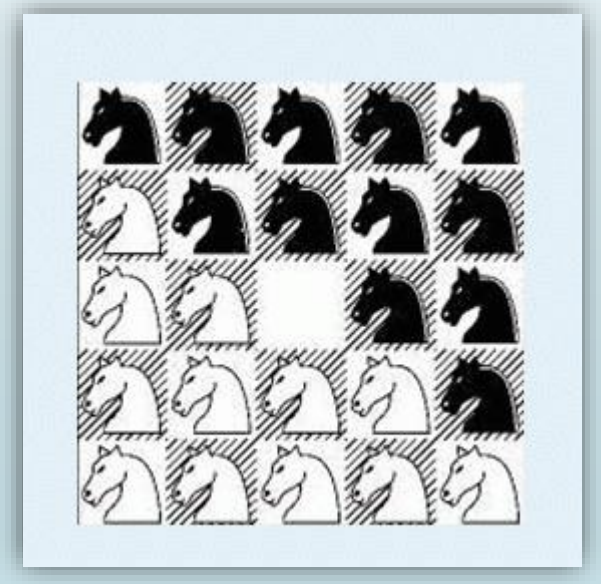
迭代+启发式！

IDA*搜索

- 在迭代加深搜索时，我们使用的深搜存在缺陷
- IDA*实际上是加了一个剪枝
  - 因为启发函数值不大于真实值，所以如果启发函数值大于 lim，就可以剪掉
- 这里的关键在于启发函数的设定

T4 「BZOJ1085」[SCOI2005] 骑士精神

- Description:
  - 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，怎样才能经过最少的移动变成如下目标棋盘（如果步数大于15步，输出 -1）：



T3 「BZOJ1085」[SCOI2005] 骑士精神

- 设置启发函数为当前棋盘和目标棋盘对应位置颜色不相同的骑士数目.
- 每次增加最大移动步数.
- dfs 时用启发函数进行估计剪枝.
- 这样就完成了这道题.

完结撒花~