

搜索



# KICKSTART 2019

- 有一个 $N \times M$ 的矩形，每个位置有一个信号站或者是空地。
- 你可以添加一个信号站，使得整个图上每个点到信号站的距离（曼哈顿距离）都小于等于 $K$ ，求出最小的这个 $K$
- $N, M \leq 250$



# USACO 09NOV LIGHTS

- 牛棚中一共有 $N$  ( $1 \leq N \leq 35$ ) 盏灯，编号为1到 $N$ 。这些灯被置于一个非常复杂的网络之中。有 $M$  ( $1 \leq M \leq 595$ ) 条很神奇的无向边，每条边连接两盏灯。每盏灯上面都带有一个开关。当按下某一盏灯的开关的时候，这盏灯本身，还有所有有边连向这盏灯的灯的状态都会被改变。
- 状态改变指的是：当一盏灯是开著的时候，这盏灯被关掉；当一盏灯是关著的时候，这盏灯被打开。问最少要按下多少个开关，才能把所有的灯都给重新打开。
- 数据保证至少有一种按开关的方案，使得所有的灯都被重新打开。



# USACO 09NOV LIGHTS

- 预处理一半的情况，HASH记录结果
- 搜索另一半的状态，判断是否存在另一半的状态使得两者有解
- 条件是：整个枚举空间分开搜索之后不会互相影响



# USACO 09NOV LIGHTS

- 预处理一半的情况，HASH记录结果
- 搜索另一半的状态，判断是否存在另一半的状态使得两者有解
- 条件是：整个枚举空间分开搜索之后不会互相影响
- 类似的想法：双向BFS & 大步小步算法



## HKU 3567 EIGHT II

- 一个 $3 \times 3$ 矩阵，由8个数字以及一个X组成，X每次可以与它相邻的数字交换，
- 给定一个开始序列以及一个目标序列，问最少需要多少次操作[U, L, R, D]能从开始序列到达目标序列，若存在多个最短路径，则输出字典序最小的一条。
- 多组数据  $T \leq 200$



# HKU 3567 EIGHT II

- HASH : 康拓展开
- 一个全排列到一个自然数的双射, 实质是当前排列在全排列中的顺序
- $X = a[n] * (n-1)! + a[n-1] * (n-2)! + \dots + a[1] * 0!$



# HKU 3567 EIGHT II

- 八数码问题:
  - BFS
  - 双向BFS
  - $A^*$ 
    - 估价函数  $f(n) = g(n) + h(n)$
    - $g$  表示从出发点到 $n$ 状态的代价
    - $h$  表示从 $n$ 状态到终点的估计代价



# HKU 3567 EIGHT II

- 如何处理字典序最小?
- $A^*$  只能搜索可行解
- 双向BFS 怎么保证字典序最小?



## HDU 3533 ESCAPE

- 有一个人要从 $(0, 0)$ 走到  $(n, m)$  , 只有 $d$ 点能量, 一秒消耗一点。
- 图中有 $k$ 个碉堡, 每个碉堡 $(x_i, y_i)$ 可以向某个固定的方向【上下左右】每隔 $t_i$ 秒放一次炮, 炮弹不能穿越另一个碉堡, 会被阻挡。人在移动的过程中不会被炮弹打到, 也就是说只有炮弹和人在某个整数时间待在同一个位置人才会被打死。
- 问人能否到达  $(n, m)$  , 如果可以输出最少消耗的能量。
- $2 \leq m, n \leq 100, 0 \leq k \leq 100, m + n \leq d \leq 1000$



# HDU 3533 ESCAPE

- BFS
- 剪枝：
  - 可行性剪枝：当前状态不可行 【剪枝】
  - 最优性剪枝：当前搜索代价+去往终点的最优代价  $\geq$  当前最优解 【剪枝】



# HDU 3533 ESCAPE

- BFS
- 剪枝：
  - 可行性剪枝：当前状态不可行 【剪枝】
  - 最优性剪枝：当前搜索代价+去往终点的最优代价  $\geq$  当前最优解 【剪枝】
- 预处理：shot[time][x][y] time时刻(x,y)是否会被攻击到。



# 埃及分数

- 在古埃及，人们使用单位分数的和（即 $1/a$ ,  $a$ 是自然数）来表示一切有理数，例如， $2/3=1/2+1/6$ ，但是不允许 $2/3=1/3+1/3$ ，因为在加数中不允许有相同的。
- 对于一个分数 $a/b$ ，表示的方法有很多种，其中加数少的比加数多的好，如果加数个数相同，那么最小的分数越大越好。
- 输入两个整数 $a, b(0 < a < b < 500)$ ，试编程计算最佳表达式。



# 埃及分数

- 迭代加深：不断加深DFS的可行层数
- IDA\* 当前状态 + 估算的最优目标数 > 层数 剪枝
- 当前分数  $a/b$  当前层数  $d$  目标层数  $D$
- 剪枝：  $a/b > (D - d) * [b/a]$
- 从  $b/a$  开始枚举分母



# 埃及分数

- 迭代加深：不断加深DFS的可行层数
- IDA\* 当前状态 + 估算的最优目标数  $>$  层数 剪枝
- 注意到这里要求最小分数最大，所以IDA\*需要在找到结果（D层）之后搜索整个搜索树（深度为D）



# 爬山 模拟退火

- 爬山算法：每次从当前解附件找到一个更优解作为当前解，直到达到（局部）最优解
  - 陷入局部最优解
  - 在平地上无法移动
- 模拟退火算法：在搜索到局部最优解后，会以一定的概率接受到另一个点的移动。也许经过几次这样的不是局部最优的移动后跳出了局部最大值A。概率会随着迭代加深逐渐降低。



# 模拟退火

```
1  T = 一个较大的值
2  while( T > T_min ) // T_min 一个较小的阈值
3  {
4      dE = J( Y(i+1) ) - J( Y(i) ) ; // 计算当前状态和最优状态的差
5      if ( dE >=0 ) //表达移动后得到更优解, 则总是接受移动
6          Y(i+1) = Y(i) ; //接受从Y(i)到Y(i+1)的移动
7      else
8      {
9          // 函数exp( dE/T )的取值范围是(0,1) , dE/T越大, 则exp( dE/T )也
10         if ( exp( dE/T ) > random( 0 , 1 ) )
11             Y(i+1) = Y(i) ; //接受从Y(i)到Y(i+1)的移动
12     }
13     T = r * T ; //降温退火 , 0<r<1 。 r越大, 降温越慢; r越小, 降温越快
14     i ++ ;
15 }
```



## CODEVS 1344

- 有  $N$  ( $\leq 20$ ) 台 PC 放在机房内，现在要求由你选定一台 PC，用共  $N-1$  条网线从这台机器开始一台接一台地依次连接他们，最后接到哪个以及连接的顺序也是由你选定的，为了节省材料，网线都拉直。求最少需要一次性购买多长的网线。（说白了，就是找出  $N$  的一个排列  $P_1 P_2 P_3 \dots P_N$  然后  $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow \dots \rightarrow P_N$  找出  $|P_1 P_2| + |P_2 P_3| + \dots + |P_{N-1} P_N|$  长度的最小值)



# CODEVS 1344

- 巧用随机化算法
  - 随机枚举MAXN个全排列
  - 模拟退火： 每次随机两个位置交换



# THE END

- 1. 巧用随机化、搜索剪枝
- 2. 注意部分分，如果想不出来完整算法就去把能做的部分分拿到
- 3. 对拍？（其实如果是暴力的话 没必要对拍吧？。。。）
- 4. 根据数据范围思考解法
- 5. 祝愿大家SHOI取的好成绩