

Solution for Contest 6

By Chancellor

Outline

- Alice[博弈搜索]** by kotori
- Bob[归纳]* by Bobgy
- Cat[线段树套树状数组]*** by Snail
- Dog[组合计数]***** by aaronzfq
- Edward[dp]* by edward_mj
- Fancy[模拟]* by FJYsmall
- Gadil[bfs]** luyi

Alice

- 可以用 $f[hp_f][cf][hp_y][cy][fst]$ 表示一个状态（状态取在两人的回合之间），进行根据博弈论原理进行dp，但主要问题就在于和局，赵云屯牌，张飞不断杀的话就会死循环。
- 但是我们可以推导出赵云的策略。
- 边界：刚过某一方的回合时，他剩1血，最多有1的手牌（刚丢完牌），接着他的对手行动时，无论是张飞还是赵云都可以直接杀死。
- 在非边界条件到边界条件的转换中，一张牌只要打出来，价值就是1点血，无论是杀对方1点还是保留自己1点。
- 1. 张飞可以预见到自己下一轮会死，就提前把手牌全部打出。2. 赵云被杀死前一定会把所有手牌用来闪。
- 张飞每轮可以把会浪费的手牌全部打掉，但是赵云每轮最多出1张，如果张飞不杀，赵云的手牌就会超出上限而不得不丢掉，产生浪费。
- 因而问题转化为研究赵云是怎么浪费的。
- 在双方进行 $2*n$ 回合后，双方都不可能死的话，一定是浪费最少的决策最优，那么赵云一定是选择每一轮都出杀，这样肯定浪费最少。
- 把双方2回合内会死的情况手算出来作为边界条件，然后以赵云必须杀作为最优决策就可以避免死循环而用dp转移了。

Alice

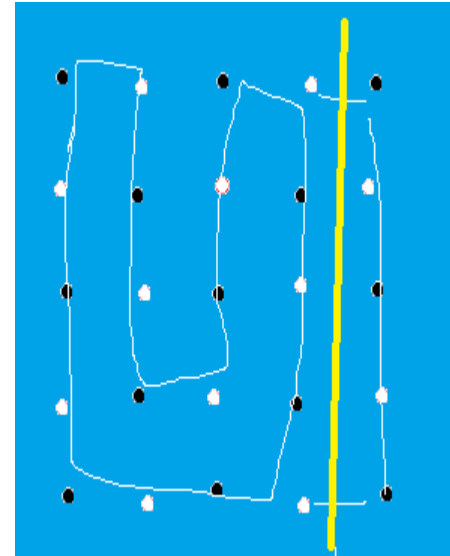
- $f[hpf][hpy][cf][cy][fst]=$
- $go(hpf-1,hpy,cf,\min(hpy,cy+1),3-fst);$
- $res=best(go(hpf,hpy-\max(0,k-cy),\min(cf+2-k,hpf),\max(0,cy-k),3-fst))$
- $(k \geq 0 \ \&\& \ k \leq cf+2)$

Solar Panel

- 先考虑边界情况，对于 $n==1 || m==1$ 时，就算把所有线都连起来也没用，断掉一根肯定就断开了。
- 但是， $n==1 \&\& m==1$ 是特例，就一个方格天生就是联通的，在底下的hint那里就是为了提醒这个。
- 对于 $n>1 \&\& m>1$ ，随便画个图看看，就会觉得用个环连起来最好。
- 这是可以证明的：
 1. 每个点最少要连2条边（否则断掉唯一的那一根它就断开了）
 2. 环（哈密尔顿回路）的长度是 $n*m$
 3. 在每个点只连2条边的时候，因为每条边被两个点共享，总长度是 $n*m*2/2=n*m$
- 所以只要存在环，环就一定是最优的。

Solar Panel

- 接下来可以构造出仅多一条边的解法：
- 如右图把一列单独分出来，剩下的部分有 $n*(m-1)$ 个点，可以用刚才方法构造出来的，最右边一列上下连起来再和左边连起来，总共花费 $n*(m-1)+n+1=n*m+1$ 条边，就是最优解了。



Cat: Beauty Pageant

- 题目大意：维护一个数据结构，支持增删固定的 n 个点、查询两个坐标都大于当前点的元素的个数和现在是否存在两个坐标都是唯一最大值的点。

Cat: Beauty Pageant

- 询问当前是否有唯一最大值的点明显可以独立做，用STL的set和map就可以。
- 剩下的询问可以离散每个点，然后以坐标 x 为下标建立线段树，在线段树的每个区间里开两个vector，分别为按pair(y , point_index) sort以后的vector和树状数组用的vector

Cat: Beauty Pageant

- 树状数组的作用在于表达第一个vector里，前 i 个元素有多少个数存在的。
- 这样每次询问线段树里面一个区间，只需要一次二分和一次树状数组的操作就可以了。
- 增加或者删除元素就直接是线段树里的单点修改，线段树里途经的结点对应修改一下树状数组即可。

Cat: Beauty Pageant

- 时间复杂度 $O((n+Q) * \lg^2 n)$
- 空间复杂度 $O(n \lg n)$

Momo: Ant's Plan

by Aldancer

NP means no perfect

题目大意

- 有一个无限的网格平面，上面有 n ($1 \leq n \leq 10$) 只蚂蚁，第 i 只蚂蚁的位置为 (x_i, y_i) 。每只蚂蚁都需要走 m ($1 \leq m \leq 2000$) 步，在走完 m 步之后，若某两只蚂蚁位于同一个格子，那么他们俩属于一个group。问每只蚂蚁走完 m 步之后，恰好形成两个group的走法。

分析：

- 变换坐标系 $(x, y) \rightarrow (x+y, x-y)$ ，俗称旋转45度。可以证明每个 (x, y) 对应唯一一个 $(x+y, x-y)$ ，反之亦然。
- 设 $tx=x+y$, $ty=x-y$ ；则变换后， $(x+1, y) \rightarrow (tx+1, ty+1)$ ， $(x-1, y) \rightarrow (tx-1, ty-1)$ ， $(x, y+1) \rightarrow (tx+1, ty-1)$ ， $(x, y-1) \rightarrow (tx-1, ty+1)$ 。那么对于四个方向的变换，原始坐标系只有一个坐标在变，而变换后的坐标系却是两个坐标一起变；这意味着，对于原始坐标系中从 (x, y) 移动到 (x', y') 的过程，在新坐标系中我们可以把 tx 和 ty 分离开独立的去处理。

分析：

- 坐标系变换后，从 (tx, ty) 经过 m 次移动变为 (tx', ty') 的方案总数[简写为： $(tx, ty) \rightarrow (tx', ty')$]可以写成 $(tx \rightarrow tx') * (ty \rightarrow ty')$
- 从 tx 经过 m 次移动变为 tx' 的方案总数计算方法：有效移动次数为 $|tx' - tx|$ ，设剩余移动次数为 $d = m - |tx' - tx|$ ，则 d 必须为偶数。其中有 $d/2$ 次正向移动和 $d/2$ 次反向移动，所以方案总数为 $C(m, d/2)$ 。
- ty 的计算过程与 tx 一致

优化

- $C(m, i) = C(m, i-1) * (m-i+1) / i$
- 对于枚举目标点:

```
for(int tz = z[0]-m; tz <= z[0]+m; tz++)
```

可以改为:

```
for(int tz = z[0]-m; tz <= z[0]+m; tz += 2)
```

Edward: Salary Increasing

题目大意：给定 n 个员工的初始工资和 m 次加薪操作。每次加薪操作将工资范围在 $[l, r]$ 间的员工的工资 $+c$ 。保证 $r_i < l_{i+1}$ 。

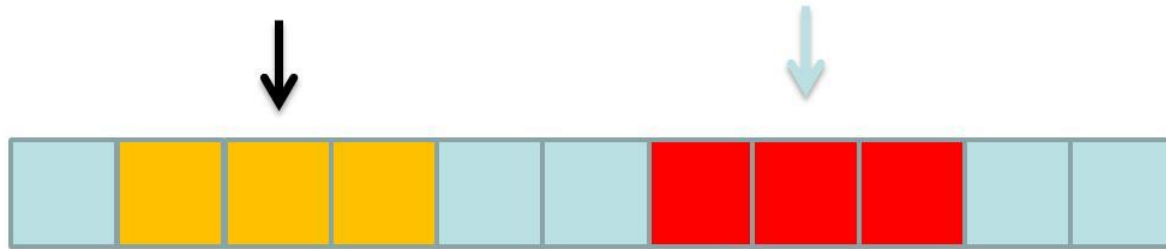
数据范围都在 10^5 级别。

Edward: Salary Increasing

- 我真的想送分.....T_T
- 给定的加薪区间不相交，那么只有两种情况。
- 第一种：本次加薪以后，工资水平进入之后的加薪区间继续加薪。
- 第二种：本次加薪以后，工资水平还在本次加薪区间或者未进入到之后的加薪区间。
- 令 $f[i]$ 表示初始薪水为 i 的人最后薪水是 $f[i]$

Edward: Salary Increasing

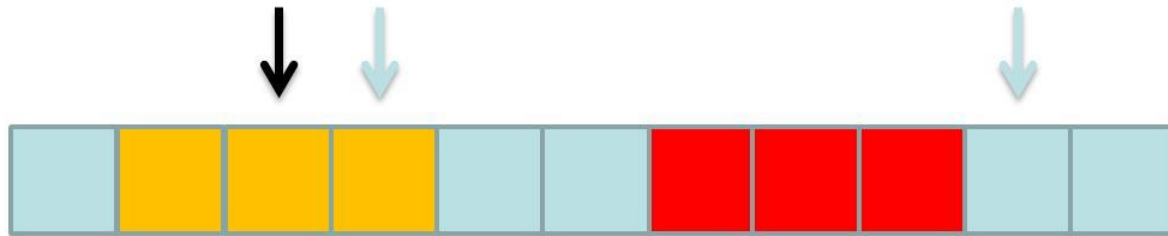
- 第一种：转移



- $f[i] = f[i + c]$

Edward: Salary Increasing

- 第二种：终止



- $f[i] = i + c$
- 然后输出 $\sum f[a[i]]$ ，注意用long long即可。

日历(Calendar)

【题目大意】

给出某年某日的一个 $6 * 7$ 日历矩阵，求 N 个月之后的日历矩阵，要求每个月的1号要出现在第一行。



【解法】

(1) 模拟。本题的数据量 $T*2000*30$ ，暴力完全能够通过。

(2) 计算每个月经过的天数，确定每个月一日的位罝。

(3) 注意到本题的日历是同生活一致，于是可以用蔡勒公式，给定某年某月某日，就能确定是星期几，直接填表

高度和当时水面高度一致

木头格子

绿色涨水高度1，不能高过最高的普通格子

蓝色降水高度1，不能低过最低的普通格子

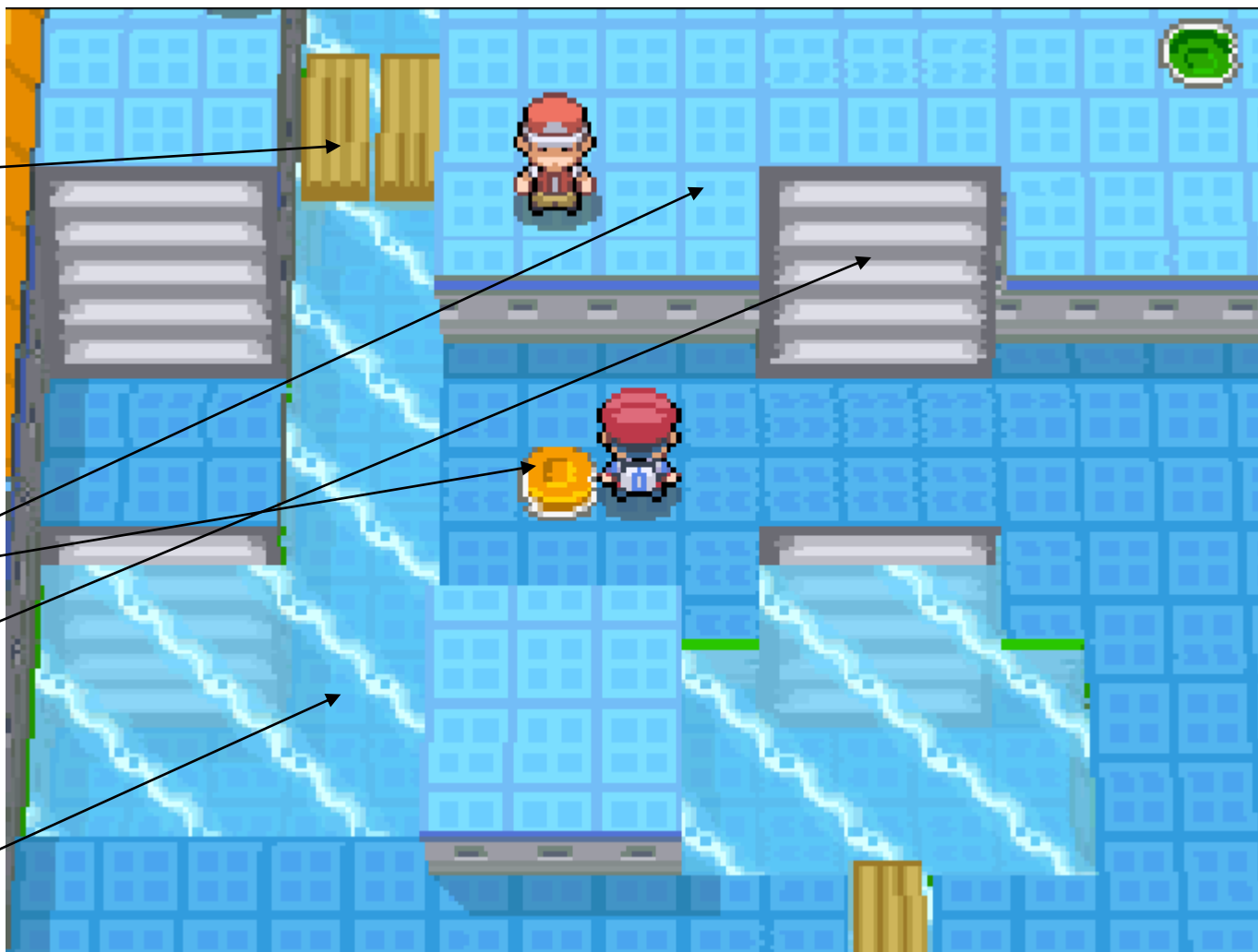
普通格子

梯子格子

可以使人所在的高度变1

水

任何时候人都不能碰到水



给一个迷宫，求从起点到终点的最短路

- 解法：就是普通的BFS， $f[i][j][l]$ 表示水面高度为 l 时人在 (i,j) 的最短路程
- 直接bfs即可

注意事项

- 时刻要判断状态的合法性
- 在起始点和终点时也会触发按钮
- 高度需要统一到0-10,或者用map（时限非常松）
- 起点一开始被水淹或碰按钮后被淹也会非法。（到终点后碰按钮被淹也不行）