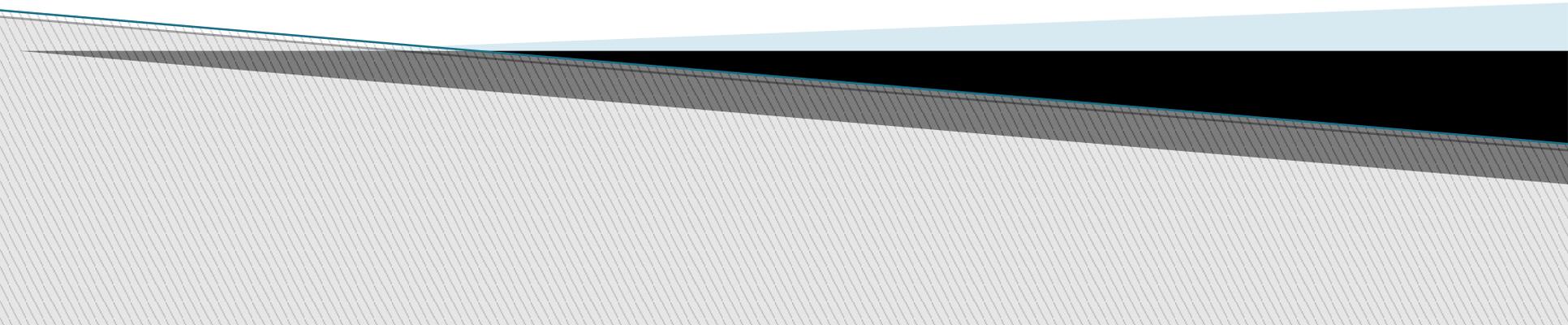


博弈论相关

李晓潇



博弈论

▶ 经济

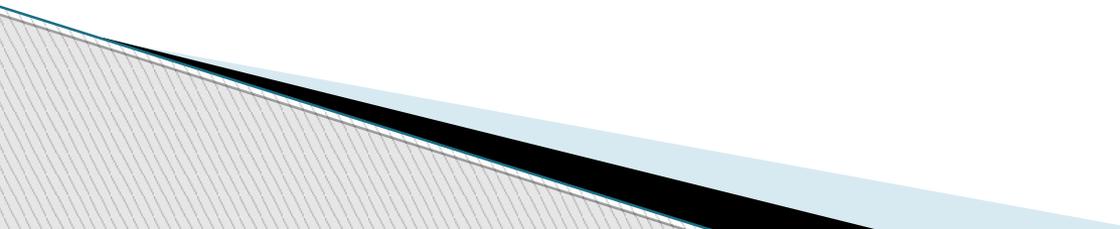
▶ 军事

▶ 竞技

▶ 爱情

▶ ○ ○ ○

恋人困境

- ▶ 有一对恋人被抓住，只有一人能活下
 - ▶ 使用锤子剪刀布来决定
 - ▶ 说好一起出锤子
 - ▶ 两人都深爱着对方
 - ▶ 但是两人都想活命
- 

恋人困境

男生	女生	锤子	剪刀	布
锤子		(0 , 0)	(1 , 0)	(-1 , 0)
剪刀		(0 , 1)	(1 , 1)	(0 , -1)
布		(0 , -1)	(-1 , 0)	(-1 , -1)

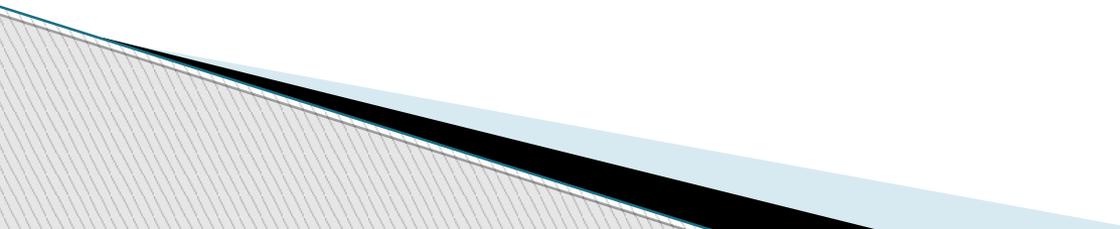
热身一下

- ▶ AB 轮流报数，规定每次报数至少为 1，最多为 m 。使得所报的数字的和到达 n 的为胜利者，求在什么情况下 A，B 有必胜策略。
- ▶ AB 轮流报数，规定每次报数至少为 1，最多为 m 。使得所报的数字的和大于等于 n 的为失败者，求在什么情况下 A，B 有必胜策略。

公平组合游戏 (ICG)

- ▶ 两名选手交替决策
- ▶ 对于某个局面，游戏的下一步选择是有限的
- ▶ 对于任意一个局面，下一步选择集合仅仅取决于局面本身，而不由其他因素影响
- ▶ 若名选手的选择集合为空，则判负；游戏本身在有限的步骤内结束

N、P 局面

- N 局面 (Next) : 将要操作的玩家有必胜策略的局面, 即先手必胜的局面
 - P 局面 (Previous) : 之前操作的玩家有必胜策略的局面, 即先手必败的局面
- 

N、P 局面求解方法

- ▶ 将所有无法移动的终止标记为 P 局面
- ▶ 若存在局面能够一步到达 P 局面，则标记为 N 局面
- ▶ 若存在局面能够到达的均为 N 局面，则标记为 P 局面
- ▶ 由于游戏在有限步能结束（即局面之间不存在环），故重复 2，3 两步必然能给所有状态打上标记（拓扑序）

小试牛刀

- ▶ AB 轮流报数，规定每次报数至少为 1，最多为 m 。使得所报的数字的和到达 n 的为胜利者，求在什么情况下 A，B 有必胜策略。
- ▶ $M = 3$

n	1	2	3	4	5	6	7	8	9
N/P	N	N	N	P	N	N	N	P	N

NIM 游戏

- ▶ 有 n 堆石子，每次可以从任意一堆取任意个数石子，两人轮流操作，无法继续判负。
- ▶ 两堆石子
 - 用 $f(i,j)$ 表示局面
- ▶ N 堆石子？
 - 分析所有 N 类状态之间的不同

一般问题

- ▶ 对于一个有向无环图，和一个棋子
- ▶ 两个选手轮流沿边移动棋子
- ▶ 无法移动判负

SG 函数

- ▶ 对于一个给定的有向无环图，定义关于图的每个顶点 x 的 SG 函数如下：
 - $g(x) = \text{mex}\{g(y) \mid y \text{ 是 } x \text{ 的后继}\}$
 - Mex 为最小的没出现过的非负整数。
- ▶ 对于一个 ICG，定义关于游戏的每个局面 x 的 SG 函数如下：
 - $g(x) = \text{mex}\{g(y) \mid y \text{ 是 } x \text{ 的后继局面}\}$
 - Mex 为最小的没出现过的非负整数。
- ▶ 将各种的 ICG 化为同样的游戏

Sprague-Grundy Theorem

- ▶ 对于任意
 - $X = X_1 + X_2 + X_3 \dots X_n$
- ▶ 有
 - $g(X) = g(X_1) \text{ xor } g(X_2) \text{ xor } g(X_3) \dots g(X_n)$

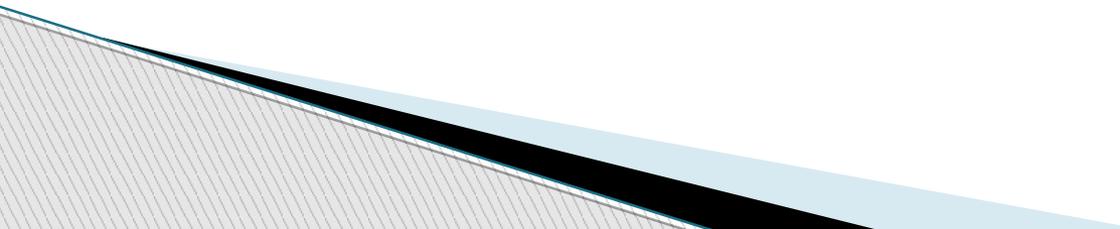
- ▶ SG 值相同局面，可以认为局面本质相同

z oj 2083

- ▶ N 条线段长短不一，AB 轮流染色，每次染色长度必须为 2 个单位长度，且每个单位长度不能被重复染色，无法操作者判负
- ▶ 考虑 $N=1$ 的情况

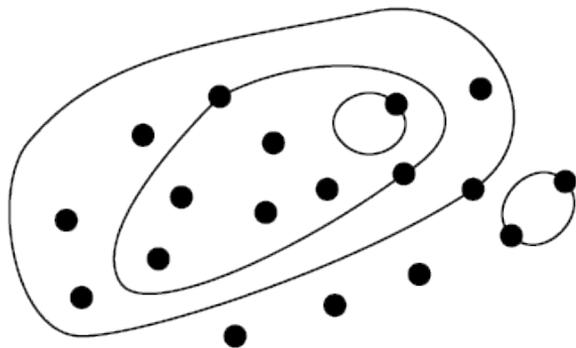
z oj 2083

- ▶ 对于每个线段当作一个子游戏

 - ▶ SG 异或和
- 

RIMS 游戏

- ▶ 在一个平面上有有限个点，和几条不相交曲线；
- ▶ 两个玩家轮流在平面上画封闭曲线
- ▶ 曲线之间不能相交，并且至少经过平面上一个点
- ▶ 无法操作则判负



RIMS 游戏

- ▶ 游戏结果仅仅与点的个数有关，与位置无关
- ▶ 画圈相当于分成两个部分，对点数求 SG 值

POJ 3710

- ▶ 给出一棵树，树上的某些节点可能结有环，任意两个环之间不相交。两人轮流操作，每次可以删掉一条边，之后去掉不与根相连通的部分。同样无法操作的判负，求谁有必胜策略？

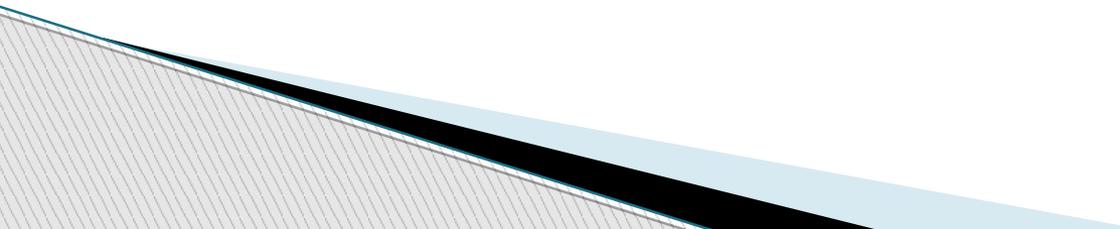
POJ 3710

- ▶ 对于一条单链
 - $G(X) =$ 链的长度
- ▶ 一棵树
 - 拆分成若干棵子树

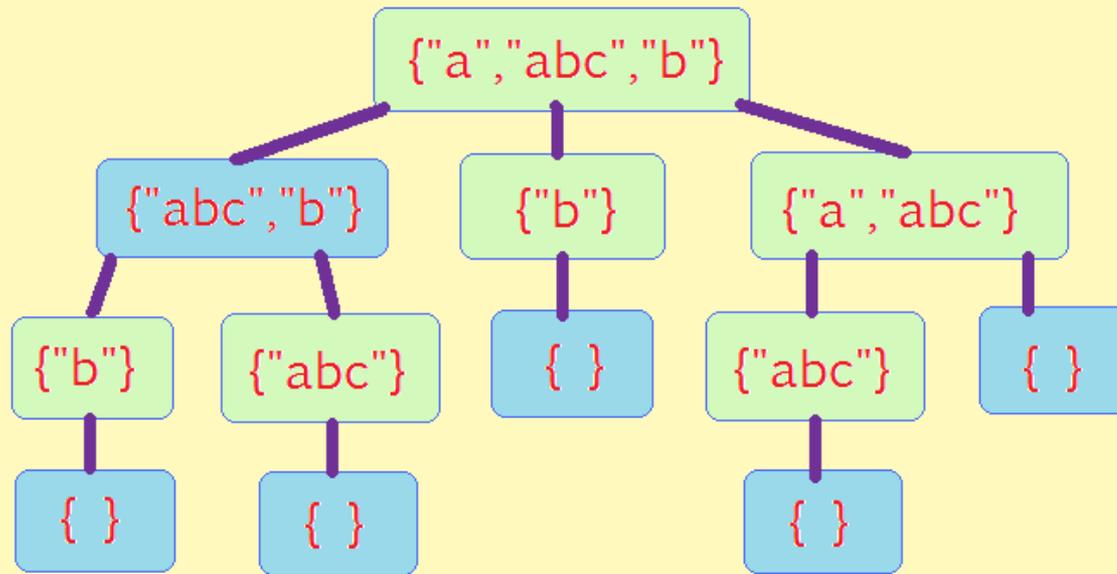
POJ 3710

- ▶ 对于环
 - 考虑环删除一条边
 - 奇数换 1，偶数环 0
- ▶ 回到原问题
 - 将环用长度为 1 和 0 的链代替
 - 求树的 SG 值
- ▶ 思考：若给出的是一张普通的无向图

WordCraft

- ▶ 游戏在一个字符串集上进行
 - ▶ 双方轮流进行
 - ▶ 选择 D 中的一个串 s ，将 D 中所有 s 的前缀（包含 s ）删除
 - ▶ 无法操作则判负
- 

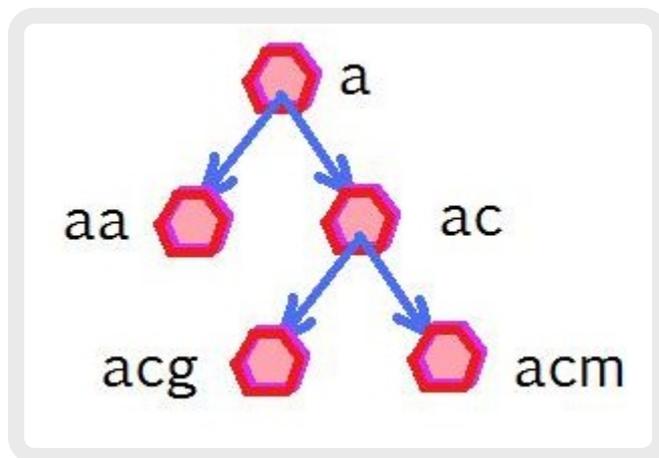
WordCraft



- ▶ 状态过多： $O(2^N)$

分析

- ▶ 为何状态过多？
 - 因为没有充分利用题目条件！
- ▶ 题目中的关键字：“前缀”。
- ▶ 这让我们想到了树结构。
- ▶ 例 :{“a” , “ aa” , “ ac” , “ acg” , “ acm”}

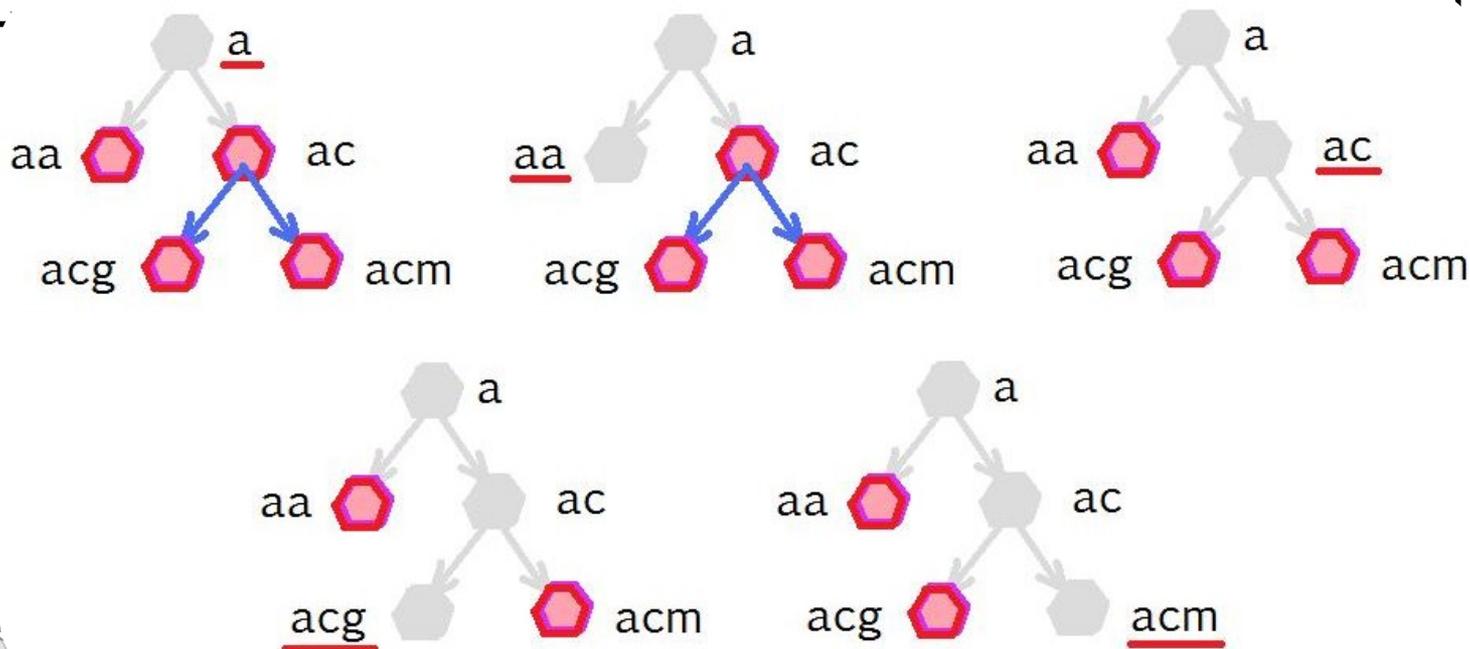


分析

- 这样，每次操作就变成：
选择某一结点，删除该结点到根的路径上的所有节点

。

- 比如



分析

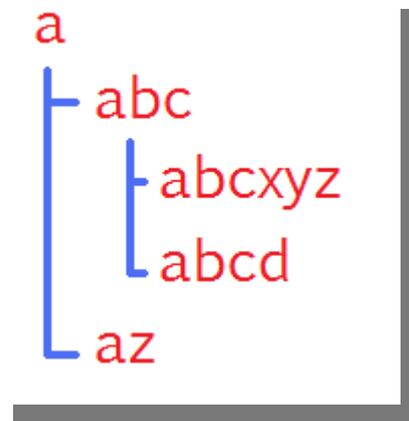
- ▶ 这样，一个局面可以看作是若干个树组成。
- ▶ 注意到所有操作只对其中的一棵树进行，与其他树无关。
- ▶ 利用 SG 函数来求解。

建树

- ▶ 需要建的是什么树？
 - 每个结点的父亲是它的（除了它自己外）最长的前缀。
- ▶ 最基本算法：
 - $O(N^2 \maxLen)$ ：对于每个结点，遍历集合 D，找到这样的父亲。
- ▶ 怎么优化？
 - 利用字典序

建树

- 每个字符串的父亲如何找？
- 不妨令 $D[0] = ""$ ；
- `void Find_Father(int i)`
- {
 `int j = i - 1;`
 `while(D[j] 不是 D[i] 的前缀)`
 `j = father[j];`
 `father[i] = j;`
}



建树

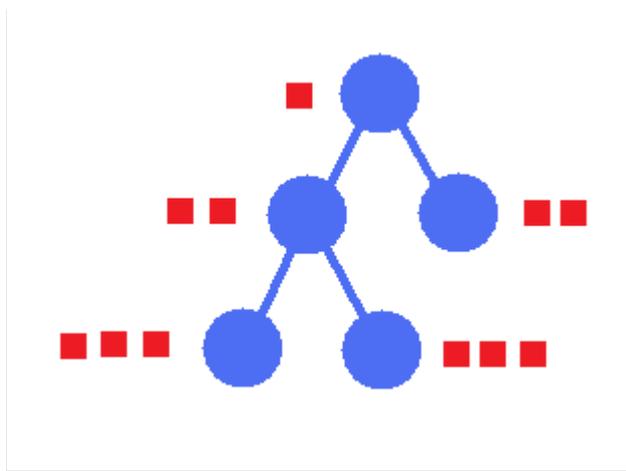
- ▶ 复杂度是多少？
 - $j = \text{father}[j]$ 最多进行 maxLen 次。
 - 每次判断前缀关系要 $O(\text{maxLen})$
 - 一共是 $O(N\text{maxLen}^2)$
- ▶ 优化？
 - 判断前缀关系的优化：
 - 计算 $D[i]$ 与 $D[i-1]$ 的公共前缀长度 comLen 即可。
 - 之后， $D[j]$ 是 $D[i]$ 的前缀当且仅当 $D[j].\text{length()} \leq \text{comLen}$
 - 这样，复杂度是 $O(N\text{maxLen})$

求解 SG 值

- ▶ 从底向上依次求解。
- ▶ 在求解某结点 i 时：
 - 枚举所有可能的操作 $O(N)$
 - 计算操走后的局面的 SG 值 $O(N)$
 - ▣ 需要预处理：每个结点的所有儿子的 SG 值 NIM 和。(xor)
 - 所以，总复杂度是 $O(N^3)$?
 - 枚举所有操作时进行一次 DFS，遍历时顺带计算出 SG 值。
 - 时间复杂度 $O(N^2)$

求解 SG 值

- ▶ 注意到这样的事实：
 - 一个结点 i 为根的子树，一次操作后局面的 SG 值不可能大于该子树的结点数目。
- ▶ 重新计算复杂度：



- $O(N \maxLen)$

课间休息 ~ ~



战火

长空

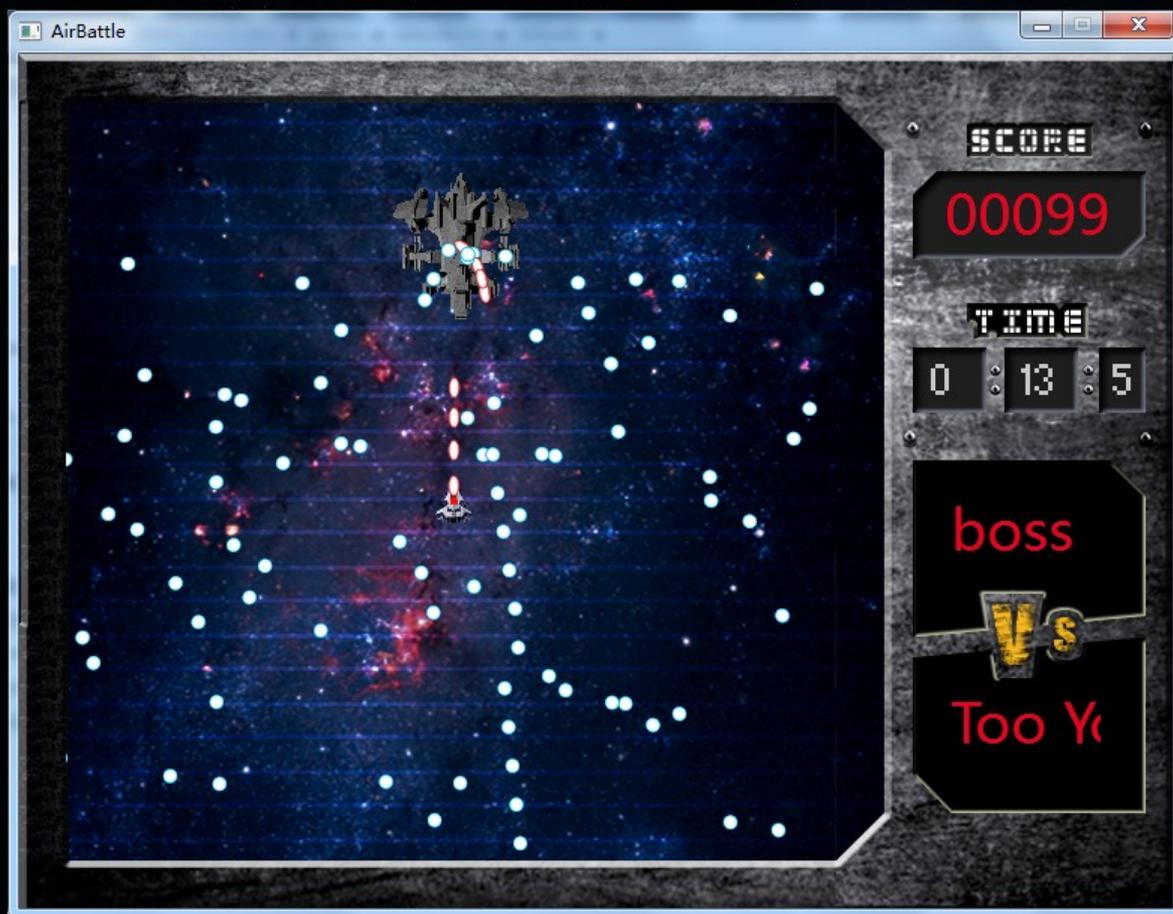


战火

长空



游戏规则



游戏机制

- 即时策略游戏 = 间隔很小的回合制游戏
- 0.1s 为一个小回合
- 最多 3000 回合 = 5 分钟

Boss 机制

- 不可移动
- 从固定点发射 5 种圆形子弹

子弹类型	1	2	3	4	5
子弹速度 (每 0.1s 移动的距离)	34	38	42	46	50
子弹半径	15	15	15	15	15
数目上限	$[20f(t)]$	$[18f(t)]$	$[16f(t)]$	$[14f(t)]$	$[12f(t)]$

- $f(t) = e^{0.00053648t}$

t	0	1292	2048	2585	3000
$f(t)$	1	2	3	4	5

Boss 机制



每回合发射每类子弹不超过
 $\max(0.4 * \text{剩余可发子弹数}, 6)$

例如：

屏幕上有第 1 类子弹 5 个

而当前限制屏幕上第 1 类子弹最多 20 个

故当前回合可以发射 1 类子弹：

$\max(0.4 * (20 - 5), 6) = 6$ 个

Plane 机制

- 匀速直线移动，速度有一个上限 40
- 静止时每回合向 Boss 发射一枚子弹
- 技能
 - 每 100 回合获得一个技能点
 - “加速”
 - 消耗 2 个技能点
 - 速度上限变为 80，持续 30 回合
 - “清屏”
 - 消耗 5 个技能点
 - 瞬间清楚屏幕上所有子弹

胜负机制

- 飞机打飞机
- 轮流扮演 Boss 和 Plane
- 比较得分

游戏时间

游戏策略

- Plane 策略
 - 我要得分，能动就不动
 - 要死啦，随机 100 方向，可以跑就跑
 - 居然跑不了，我放大招
 - 我擦。。没有技能点。。再见

游戏策略

- Boss 策略
 - 我 RP 好，我随便打
 - 瞄着你，追你到天涯海角
 - 还是构造个 nb 弹幕？

讨论时间

Boss 想法

- 封杀小飞机 VS 逼小飞机走位
- 加长型 VS 密不透风型

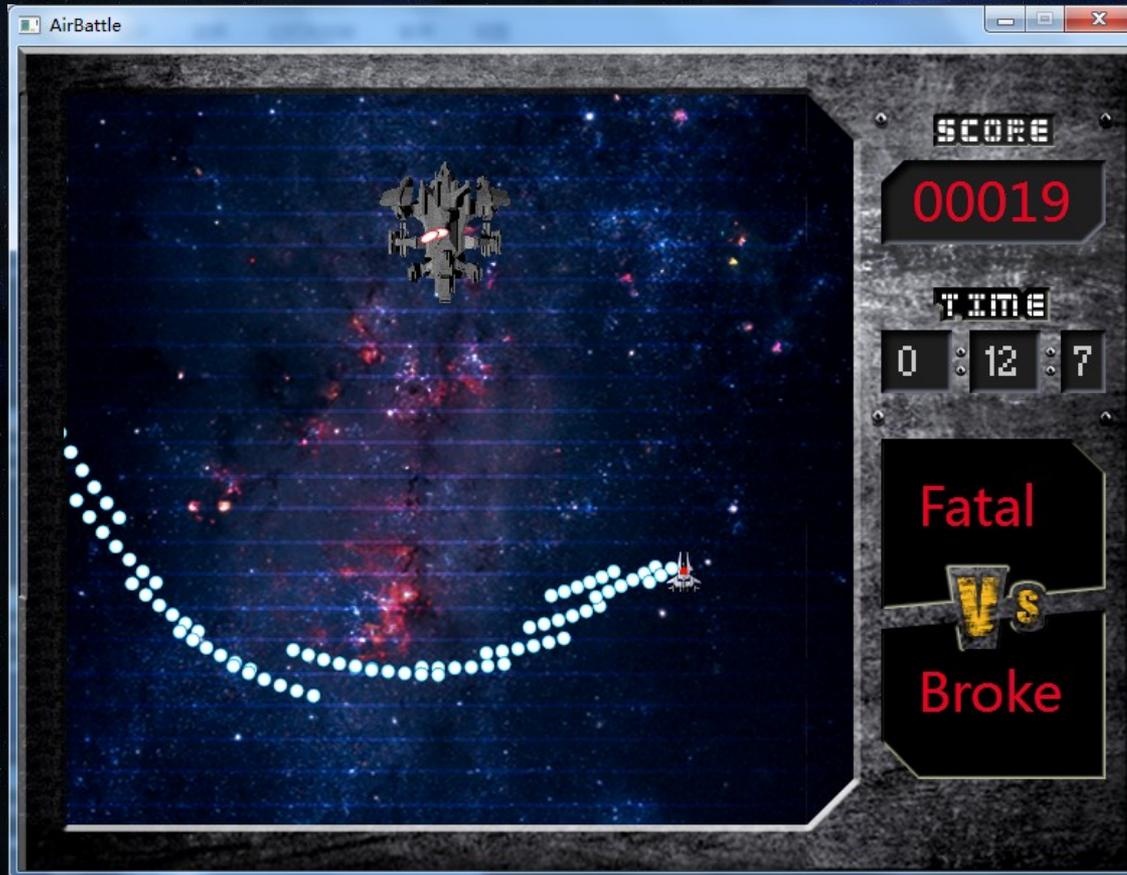
Plane 想法

- 地图安全区域
 - 靠边 VS 居中
- 寻找可行路径
 - 限定范围
 - R^2 VS 离散化
- 技能的使用
 - 加速 VS 清屏

策略之大 boss 篇

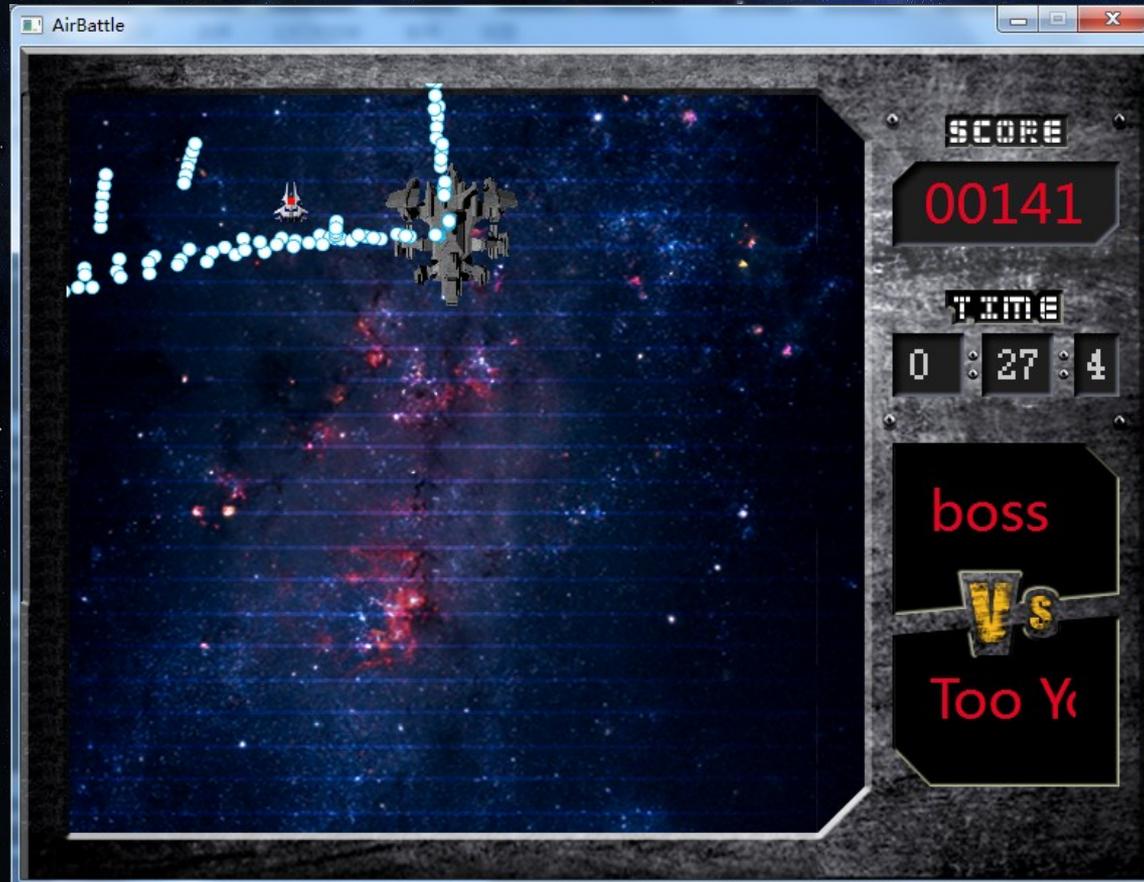
扇形弹幕

- 利用子弹之间的速度差



条形弹幕

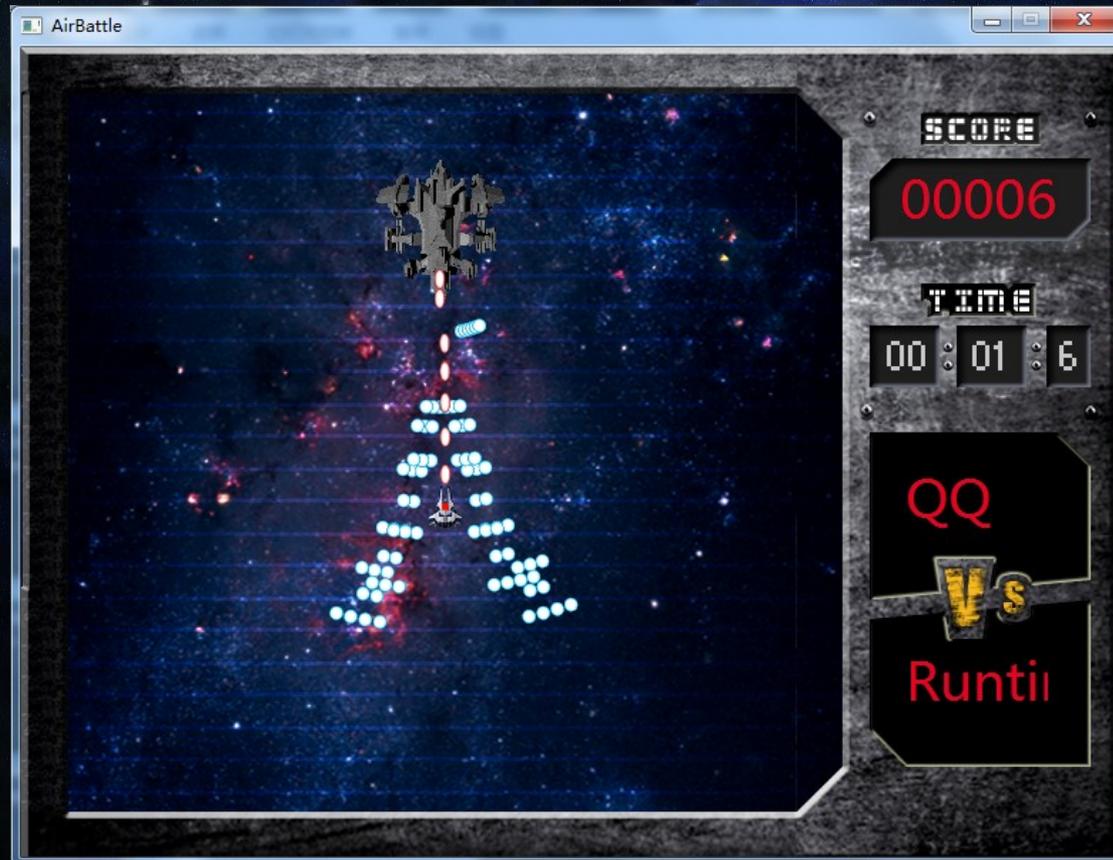
- 利用飞机的麻痹大意，地图不对称



策略之小飞机篇

裤衩弹幕

- 欺骗小飞机



思路一

- 特判弹幕类型，写出针对的 AI
- 圆弧：计算夹角
- 直线：寻找缝隙向中间移动
- 裤衩：躲开子弹密集区域
- 其他：.....

思路二：动态规划

- 离散化将平面分成 $n*m$ 的网格
- $F[i][j][k]$ 考虑之后 ($T < 27$) 个回合，若第 k 个回合在 (i, j) 点，那么到第 T 回合最多能等多少分
- 求 $F[\text{now}_i][\text{now}_j][0]$

状态转移方程

$$F[i][j][k] = \text{Max} \begin{cases} F[i][j][k+1] + 1 & \text{停留是安全的} \\ F[i'][j'][k+1] & \text{向 } (i',j') \text{ 移动合法且安全} \\ \text{全} \end{cases}$$

边界条件： $F[i][j][T] = 0$

时间复杂度

- $O(n * m * T * \text{相距 } 40 \text{ 以内格子数} * \text{子弹数目})$
- 设 $n = 50, m = 50, T = 20$
- 每回合计算次数： 250000000

算法优化

- 限制移动的方向数
- 记忆化搜索
 - 消除无用状态
- 估价函数剪枝
 - 追求效率和得分率的平衡
- 子弹判定方法
 - 避免重复判定，避免无意义的判定
- 技能的使用限制

其他小技巧

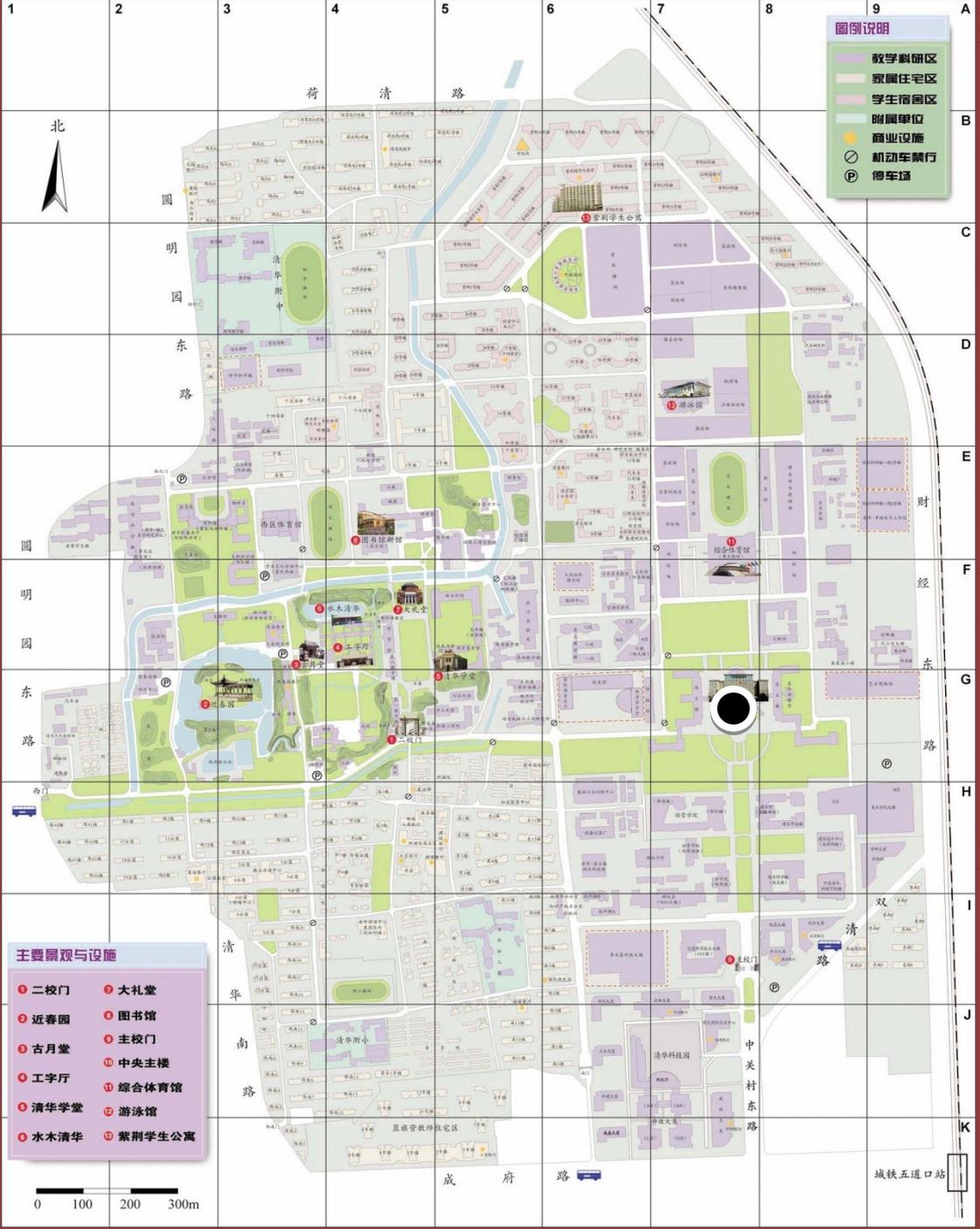
- 每次移动后尽量移动回中间位置
- 尽量在屏幕的中上方向
-

其他比赛信息

1	2	3	4	5	6	7	8
何朴藩	周奕超	李晓潇	冯齐纬	巫立凡	付钊	姜秀宝	苏蕉
9	10	11	12	13	14	15	16
李振		任印政	周昕宇	闫承稷	陈高远	林源	任杰

决赛时间

- 今晚 7 : 00
- 主楼一层报告厅 (进门正对最大的楼哦)
- 观众奖品
 - Kindle Fire
 - 遥控飞机
 - 拍立得

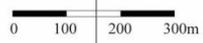


图例说明

- 教学科研区
- 家属住宅区
- 学生宿舍区
- 附属单位
- 商业设施
- 机动车骑行
- 停车场

主要景观与设施

- ① 二校门
- ② 近春园
- ③ 古月堂
- ④ 工字厅
- ⑤ 清华学堂
- ⑥ 水木清华
- ⑦ 大礼堂
- ⑧ 图书馆
- ⑨ 主校门
- ⑩ 中央主楼
- ⑪ 综合体育馆
- ⑫ 游泳馆
- ⑬ 紫荆学生公寓



城铁五道口站

谢谢 ^_^

李晓潇

lxx1991@gmail.com