



吉林大学 2022 年春季学期

《数据结构》课程设计第 5-7 次课赛道 B

不围棋游戏编程对抗赛

不围棋 (NOGO) 由围棋衍生而来, 棋盘比围棋小, 规则与围棋相反, 玩起来比围棋简单。在本题中, 你的任务是编写不围棋 AI 程序, 即让程序自动下棋, 并在北京大学 Botzone 平台 (<https://www.botzone.org.cn/>) 与其他同学对战, 进而决出冠军及名次。不围棋也是“中国大学生计算机博弈大赛 (<http://computergames.caa.cn/>)”和人工智能游戏方向的国际权威会议“IEEE Conference on Games (<https://iee-cog.org/2022/>)”的比赛项目。本题表现突出的队伍还可参与上述两项竞赛, 若取得较好成绩, 在今后求职应聘或者保研时将是一个重要砝码。

一、游戏规则与术语:

1. 棋盘为 9×9 围棋棋盘。

2. 黑子先手, 双方轮流落子, 落子后棋子不可移动。

3. “连”的定义: “连”就是棋子和棋子接在一块, 成为一个整体。横是连, 竖是连, 斜线不算连。例如图 1 中带方形的两颗棋子是相连的, 带三角形的三颗棋子是相连的, 右边的带圆形的三颗棋子不相连。

4. “气”的定义: “气”是围棋和不围棋里最核心的概念。一个棋子在棋盘上, 与它直线紧邻的空点是这个棋子的“气”。棋子直线紧邻的点上, 如果有同色棋子存在, 则它们便相互连接成一个不可分割的整体, 它们的气也应一并计算。棋子直线紧邻的点上, 如果有异色棋子存在, 这口气就不复存在。如所有的气均为对方所占据, 便呈无气状态。例如图 2 中, 角上一颗黑子有 2 口气, 其右侧上边的一颗黑子有 3 口气, 中间一颗黑子有 4 口气。右边两个白子相连有 6 口气, 下边三个白子相连有 8 口气。

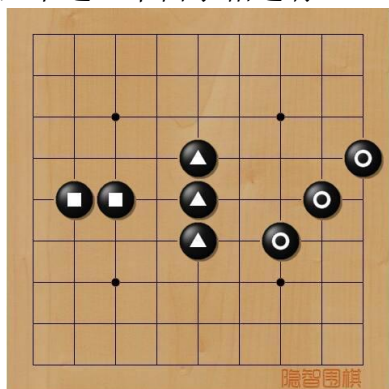


图 1 连

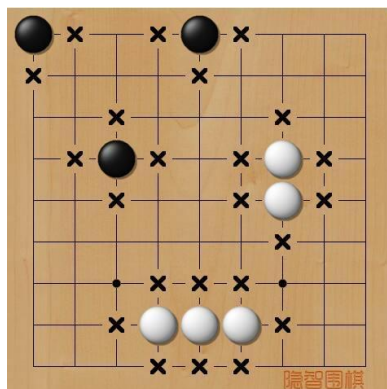


图 2 气

5. “吃子”的定义: 落子后, 使对方的棋子变成无气状态就是吃子。例如在图 3 中, 如果白棋下在 “X” 标记点上, 就会把黑棋变成无气, 把黑棋吃掉。

6. 不围棋对弈的目标不是吃掉对方的棋子，恰恰相反，如果一方落子后吃掉了对方的棋子，则落子一方判负。

7. 如果一方在棋盘上某个点落子后，该子将呈现无气状态，那么相当于自杀，例如图 4 中，白棋若在“×”处落子即为自杀。不围棋对弈禁止自杀，落子自杀一方判负。

8. 对弈禁止空手（Pass），即轮到本方走棋时，若无棋可走，则本方判负。

9. 对弈结果只有胜负，没有和棋。

10. 每步走棋限时 1 秒，超时判负。内存限定 256MB。

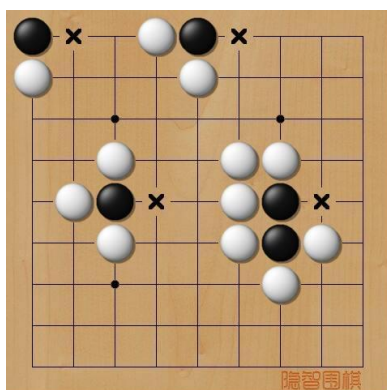


图 3 吃子

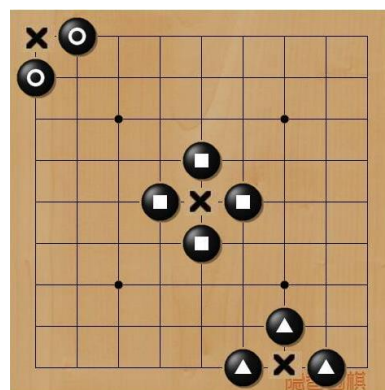


图 4 自杀

建议大家实际玩玩这款游戏，玩个十几局就掌握其规则及玩法了。PC 版已发给大家（若无法打开，可能需要安装 visual studio），手机版可在 app store 或安卓市场下载。大家也可以查阅相关网络资料，进一步了解不围棋的下棋技巧和经验。

二、代码实现：

大家无需掌握图形界面编程技术，只需将代码上传至北京大学 Botzone 对战平台（<https://www.botzone.org.cn/>），对战平台将自动进行图形化对战比赛，并给出排名。本题编程语言为 C/C++ 语言，允许使用 C++ STL。

（1）棋盘坐标设定

9×9 的二维棋盘分为横轴和纵轴两个维度，以左上角为坐标原点(0,0)。坐标系如图5所示。在程序中棋盘信息存储在数组 `int board[9][9]` 中，下标从 0 开始，元素 `board[x][y]` 有 0、1、-1 三种可能取值，分别表示棋盘(x, y)处为空白（没有任何棋子）、对方棋子、我方棋子。

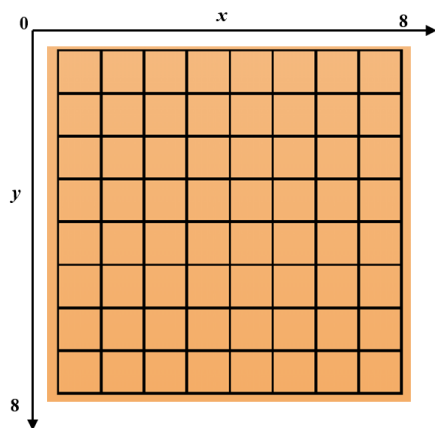


图 5 棋盘坐标系

(2) 你的程序与对战平台的交互方式

本题提供给大家**样例程序**，样例程序已经包含了程序与对战平台的交互功能，你只需在样例代码基础上开发，基于当前棋局，输出本回合的决策结果。你的程序（在botzone网站里也称为bot）的生命周期仅为一次决策：读入当前棋盘状态，输出本回合你的下棋（落子）位置。在整个对弈过程中，对战平台会多次调用你的程序（每下一步棋调用你一次，直至对弈结束）。

了解下面标注黄色高亮的内容有助于理解程序的运行原理和流程。若不想或无法理解下面内容，可跳过而直接看（3）节，对完成本题没有影响。

对战平台每次调用你的程序时：首先将之前对方和本方落子的位置作为输入传送给你的程序，从而建立起棋盘信息board数组，然后你的程序根据棋盘信息进行决策，并向对战平台输出你的决策结果，即本方的落子位置。

你的bot与对战平台有两种交互方式：简单交互和JSON交互（你可任选其一）。

➤ 简单交互

使用这种交互方式，你的程序通过一系列数字与对战平台进行交互，格式形如下表左侧所示。而你的程序就是要读取这些数字，建立棋盘信息，并基于棋盘信息做决策，最后输出第 k 回合本方落子位置。详见示例程序。例如图6的棋盘状态对应下表右侧的输入。

k	//双方已下棋的回合数	4
$x_1^{\text{对方}}$ $y_1^{\text{对方}}$	//第1回合对方落子位置	-1 -1
$x_1^{\text{本方}}$ $y_1^{\text{本方}}$	//第1回合本方落子位置	2 3
$x_2^{\text{对方}}$ $y_2^{\text{对方}}$	//第2回合对方落子位置	0 8
$x_2^{\text{本方}}$ $y_2^{\text{本方}}$	//第2回合本方落子位置	7 5
.....		6 8
		5 8
		1 4
$x_{k-1}^{\text{对方}}$ $y_{k-1}^{\text{对方}}$	//第 $k-1$ 回合对方落子位置	
$x_{k-1}^{\text{本方}}$ $y_{k-1}^{\text{本方}}$	//第 $k-1$ 回合本方落子位置	
$x_k^{\text{对方}}$ $y_k^{\text{对方}}$	//第 k 回合对方落子位置	

假设本方是黑方，对于上面的信息，可知之前的棋应该是这么下的：我方第一回合收到(-1,-1)，然后我方下棋(2,3)，然后对方下棋(0,8)，然后我方下棋(7,5)...以此类推，形成图6的棋盘。

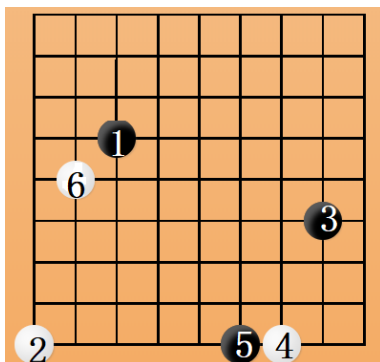


图6 对战平台与博弈程序通信示例

➤ JSON交互

你可以理解为程序通过一系列“特殊格式的字符串”与对战平台进行交互，其格式形如：

```
{"requests":[{"x":-1,"y":-1},{"x":0,"y":8},{"x":6,"y":8},{"x":1,"y":4}],
"responses":[{"x":2,"y":3},{"x":7,"y":5},{"x":5,"y":8}]}
```

其中requests后面是对方以往落子的位置，如果是黑方的第一回合，则 request 为 {"x":-1,"y":-1}。responses后面是本方以往的落子位置。你的程序可以对上述字符串进行自动解析，解析出x坐标和y坐标（样例程序已经包含了解析代码，无需你来编程解析），这样你可以根据这些x和y坐标，建立当前棋盘状态信息（样例程序已经实现该功能，并将棋盘信息存入board数组，你只需根据board数组所表示的棋盘信息，进行决策即可）。详见示例程序。

假设本方是黑方，对于上面的信息，可知之前的棋应该是这么下的：我方第一回合收到(-1,-1)，然后我方下棋(2,3)，然后对方下棋(0,8)，然后我方下棋(7,5)...以此类推，从而形成图6的棋盘。

关于bot与平台的交互方式的更详细描述可参考：

<https://wiki.botzone.org.cn/index.php?title=Bot#.E4.BA.A4.E4.BA.92>

<https://wiki.botzone.org.cn/index.php?title=NoGo>

(3) 博弈程序编写

提供给大家JSON交互和简单交互的两个样例程序，分别为nogoai.cpp和nogo_simple.cpp，里面包含了详细注释，容易阅读。代码中棋局初始化和输入输出操作你都不用管，**你只需要实现代码中下图红框部分，用你的策略替换掉红框中的样例策略。new_x和new_y就是你的决策结果。**

```
... = input["requests"][i]["x"].asInt(), y = input["requests"][i]["y"].asInt();//对方，注意此y
if (x != -1) board[x][y] = 1;
x = input["responses"][i]["x"].asInt(), y = input["responses"][i]["y"].asInt();//我方，注意
if (x != -1) board[x][y] = -1;
}
x = input["requests"][turnID]["x"].asInt(), y = input["requests"][turnID]["y"].asInt();//对方，
if (x != -1) board[x][y] = 1;
//此时board[] []里存储的就是当前棋盘的所有棋子信息，x和y存的是对方最近一步下的棋

/*****在下面填充你的代码，决策结果（本方将落子的位置）存入new_x和new_y中*****/
//下面仅为随机策略的示例代码，可删除
int available_list[81]; //合法位置表
int k = 0;
for (int i = 0; i<9; i++)
    for (int j = 0; j<9; j++)
        if (judgeAvailable(i, j, x == -1 ? 1 : -1))
        {
            available_list[k] = i * 9 + j;
            k++;
        }
int result = available_list[rand() % k];
int new_x = result / 9;    int new_y = result % 9;
/*****在上方填充你的代码，决策结果（本方将落子的位置）存入new_x和new_y中*****/

// 输出决策JSON
Json::Value ret;
Json::Value action;
action["x"] = new_x; action["y"] = new_y;
```

使用任何 IDE 均可运行样例程序，但注意 json 交互需要将 nogoai.cpp 文件与 json 文件夹、jsoncpp.cpp 文件保持在同一文件夹下，详见：

<https://wiki.botzone.org.cn/index.php?title=%E8%B0%83%E8%AF%95#.E6.9C.AC.E5.9C.B0.E8.B0.83.E8.AF.95>

（4）博弈程序的调试

➤ 本地调试

在本地编译运行程序，本地调试时可采用“简单交互”方式，并结合输入输出重定向，以数字形式作为输入，并观察程序输出结果。

➤ 在Botzone网站上在线调试

Botzone 提供了比较完善的调试功能。点击网页右上角的“我的 Bot”，然后点击下方“创建新 Bot”，如图 7。然后填写 bot 信息，上传代码文件，如图 8。请不要勾选“开源”，为避免抄袭，将对开源者扣分。**Botzone 提供了较好的版本管理功能，方便管理程序的各个版本。**若想对已有的 bot 增加新的版本，可如图 9 所示点击加号。



图 7 创建新 Bot



图 8 配置 Bot 信息上传代码



图 9 为已有 bot 增加新版本



图 10 创建游戏桌

在线调试，可回到首页，手动创建游戏桌，如图 10 所示。然后指定玩家进行对局，如图 11 所示，对局双方可以是已有的 Bot，也可以是房主（人类），即你自己。



图 11 指定玩家进行对局调试

在对局页面（也可通过右上角菜单“我的对局”选择对局点击“回放”）在线调试，点击下方“切换到调试模式”，如图 12 所示。详细调试信息可以点击右下角“log 查看工具（调试用）”查看每回合每方具体的输入和输出信息，如图 13，选择“查看完整 Log”，也可以查看程序的运行时间。为防止程序超时，可以尝试进行卡时，比如在每次迭代/循环时，通过 `clock()` 或者 `time()` 函数来判断程序运行了多久，从而决定是否提前返回。

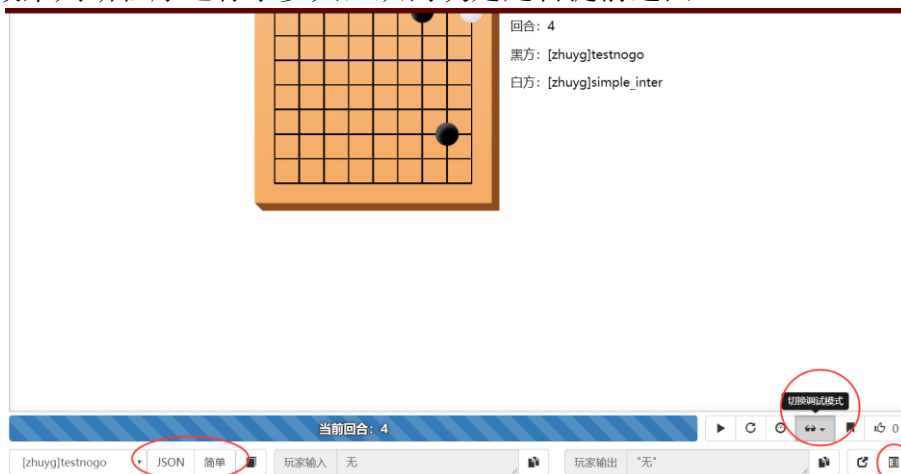


图 12



图 13

在 JSON 交互模式下，可将程序中的调试信息写入代码中的 `ret["debug"]` 处，在线调试的 log 查看工具中的“玩家输出”可显示出运行过程中的调试信息，如图 14 所示。

```
int result = available_list[rand() % K];
int new_x = result / 9; int new_y = result % 9;
/******在上方填充你的代码，决策结果（本方将落子的位置）存入new_x和new_y中*****/
/******

// 输出决策JSON
Json::Value ret;
Json::Value action;
action["x"] = new_x; action["y"] = new_y;
ret["response"] = action;
ret["debug"] = turnID; //调试信息可写在这里
Json::FastWriter writer;

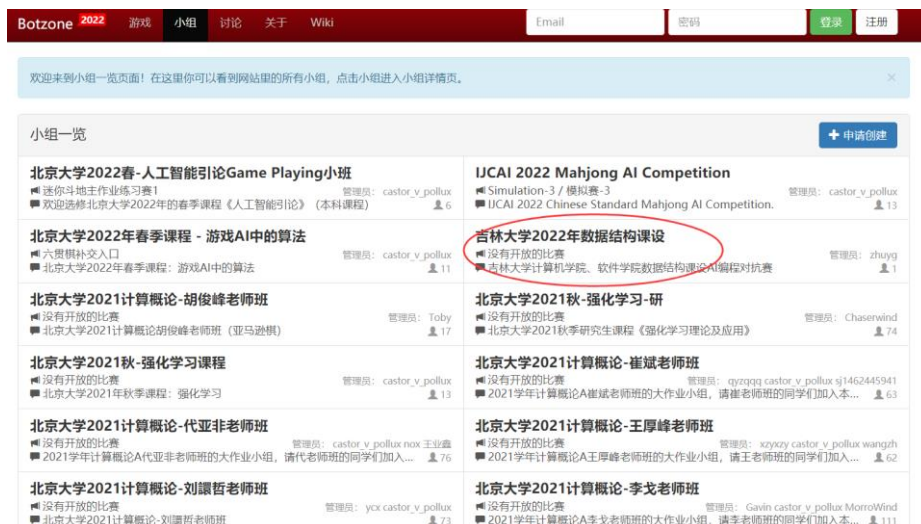
cout << writer.write(ret) << endl;
return 0;
}
```



图 14

你的程序在本地的输出和平台的输出不一样怎么办？（1）你的程序可能存在漏洞，在平台上可能会暴露出数组越界、地址越界、除零、未初始化等问题。建议仔细检查自己的代码。TScanCode 是一个很好的工具。（2）如果采用了随机数，可能你本地生成的随机数和平台生成的随机数不一致。

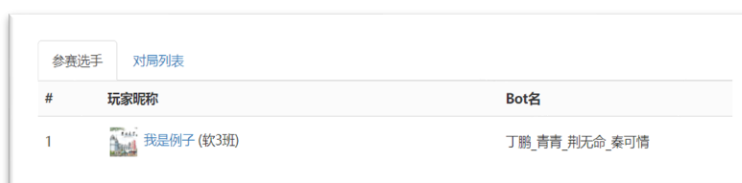
三、程序提交方式：



先通过老师提供的 <https://wj.qq.com/s2/9966473/ccb1/> 链接报名组队，在老师审核通过后，由队长注册 Botzone 账号并加入 2022 年吉大课设小组，按规定时间参加积分赛。该账号应为本组所有成员的公用账号，而非组长的私人账号。组内成员均可用该账号提交和测试代码。每组仅该账号允许加入吉大课设小组并参赛。账号和 bot 命名要求如下：

- （1）**账号昵称**设为队名，可以起一个有特色的队名，但建议名字不要过长。
- （2）**组中昵称**设为“计/软+班号+班”的格式，如“计 35 班”、“软 3 班”，不要写加号。
- （3）**Bot 名**设为“队长姓名_成员 1 姓名_成员 2 姓名_成员 3 姓名”的格式。

例如下图，账号昵称/队名为“我是例子”，组中昵称为“软 3 班”，bot 名为“丁鹏_青青_荆无命_秦可情”



未按要求命名者成绩将会受到影响。对于少数民族姓名较长的同学，由于系统限制了 bot 名长度，所以建议简写姓名，能辨别出是你即可。

程序提交截止期为第 8 次课开始前，具体时间另行公布。

四、实验报告提交方式：

提交程序后一周内，每队通过超星作业的方式提交实验报告 **PDF 版**，报告每队 1 份，仅队长提交，成员不用提交。实验报告模板另行发布。文件以“赛道号_队长姓名（班级）_成员姓名（班级）_成员姓名（班级）...”命名，例如“B_刘备（21 班）_关羽（22 班）_张飞（23 班）.pdf”，不用写学院，对于少数民族姓名较长的同学，可简写姓名，能辨别出是你即可。

请注意：必须提交 PDF 版，可在 Microsoft Word 中将 doc 文件另存为 pdf 文件。系统无法自动下载回收非 PDF 文件，不符合文件格式要求的报告将不会被老师将收到，将按 0 分记。

五、组队方法：

本题以团队形式完成，每队 1-4 人，鼓励班内或寝室内组队，也可在本学院范围内自由组队，若跨班级组队，检查程序（答辩）时应在队长所在班检查，队内其他班队员也要到场。也可由老师自由组队，希望老师自由组队的同学，可在链接中先填上自己和已有队员的名字，剩余队员由老师给你随机补齐。不鼓励单人组队，建议寻找志同道合的队友齐心协力，共同完成。若有同学实在想单人组队，也尊重这些同学的意愿。

首先通过此链接报名：<https://wj.qq.com/s2/9966473/ccb1/>

六、评测方法：

以程序对战的方式进行评测，比赛分为两阶段。

- **第一阶段：院内比赛。**本学院内所有队伍采用瑞士轮赛制进行积分赛，胜一场积 2 分，平一场积 1 分，负一场积 0 分。为保证公平性，积分赛一共进行 3 次，根据 3 次积分赛的加权总积分，得出本学院各队排名。大家在本课程的成绩将由本阶段比赛决定。
- **第二阶段：诸神之战。**计算机学院前 15 名的队伍和软件学院前 10 名的队伍进入第二阶段。若院内竞争激烈比分接近，也可能适当放宽每院出线名额。第二阶段仍进行 3 场积分赛，根据 3 场比赛总积分，决出两院总冠军。本阶段仅供娱乐，与课程成绩无关。
- **第三阶段：吉大-北大友谊赛。**将组织一场吉大前几名与 2020 级北大的前几名的友谊赛。2020 年高考时你们输给了他们，但并不意味着现在仍会输给他们，给大家提供一个再次与他们同场竞技的机会。
- **第四阶段：**表现突出的同学还可参加中国大学生计算机博弈大赛或 IEEE CoG 2022 Competition.

七、成绩评定方法：

本题满分为 200 分，比赛得分占 70 分，程序原创性得分 40 分，实验报告占 30 分，小组讨论 20 分，答辩 40 分。比赛得分基于对战排名决定，比如本学院前 x 名为 100 分，x+1 至 y 名 90 分，y+1 至 z 名 80 分……以此类推，具体细则根据最终的参与队伍数量决定。程序原创性分数由你代码或算法策略的原创程度决定，比如你的代码是完全原创的还是借鉴了网上代码或框架，你的评估函数是独创或改进的还是直接用别人现成的，你的代码和网上或其他同学代码的雷同度等等。实验报告的分数由各带班老师评定。

其中比赛得分、程序原创性得分、实验报告得分组内成员相同。例如某队这三项得分为 135 分，则队内每个队员均得到 135 分。小组讨论、答辩分数各队员不同，比如某同学没参与小组讨论，则小组讨论分为 0 分。答辩分数则根据队员的具体贡献由老师评定，如果各队员齐心协力，基本做了等同的贡献，那么队员都可获得相同的答辩分数。

简单说，本题满分 200 分，只要队内齐心协力，每名队员都可能获得 200 分，都有可能获得优秀。队内优秀名额不设上限，不会一个队最多只有一个优秀。

当然，如果发现某个队员明显划水，则老师可对其倒扣分，直至 0 分。若发现某队在分工上集体作假，比如某队员没做什么贡献，但团队却说他做了很多贡献，则全队都将视为违纪。

提交的文件没按照题目要求的格式命名的，视情节轻重进行一定扣分，尤其是 bot 名、队伍名等没按规则命名导致无法辨别出学生班级姓名的，则比赛结果无效，得分全部扣减。

如果仅提交样例程序，则无论比赛结果如何，本题均记 0 分。

我们鼓励大家创新思维，若对传统算法进行了创新性的改进，提出了原创性很高的解法，或在算法中融入了新颖的技巧、策略等，可给予加分。

以上规则也可能在课程进行中有所调整，如果同学们有好的建议，也欢迎提出来，好的建议我们会采纳。

八、前情回顾：

2018 级开展了两项全员参与的对抗赛：

比赛项目	贪吃蛇大作战	六子棋锦标赛
具体内容	编写程序控制贪吃蛇，在动态变化的场地中与对手比拼，有效躲避障碍物并合理选择道具，力争在最短的时间内尽可能多地吃掉食物。	编写六子棋 AI 程序，并与其他同学对弈。六子棋与五子棋类似，六个棋子连成一条线则赢。
对战平台	朱老师开发	中国大学生计算机博弈大赛官方平台
比赛结果	两院前 7 名中计算机学院占据 6 席，但冠军被软件学院宁玉恒同学获得，该同学采用了巧妙的算法，还专门设计了精巧的数学模型融入算法，从而横扫所有对手夺冠。该同学已保送北京航空航天大学，目前正在朱老师名下做毕设。	计算机学院包揽两院前 6 名，冠军被贾林瀚同学获得，随后老师资助该同学参加 2020 年全国大学生计算机博弈大赛，获国家二等奖。大三时该同学获得 ICPC 亚洲区域赛银奖。该同学已保送南京大学，目前也在朱老师名下做毕设。

2019 级开展了两项全员参与的对抗赛：

比赛项目	不围棋锦标赛	贪吃蛇大作战 II
具体内容	编写不围棋 AI 程序，并与其他同学对弈。不围棋由围棋衍生而来，棋盘比围棋小，且规则与围棋相反。	与去年的贪吃蛇大作战不同，本次地图中没有食物，但蛇身会自动变长。本方目标是尽可能的挤压对方的行动空间，让对方无路可走，从而逼死对方。
对战平台	Botzone	Botzone
比赛结果	两院前 7 名中计算机学院占据 6 席，但冠军被软件学院朱志放同学获得，该同学在此期间投入了巨大时间精力，就连其他课的老师都说“这个同学最近总研究围棋...”。随后老师资助其参加 2021 年全国大学生计算机博弈大赛，获全国季军。此外，两院亚军张轶博同学，本已在大二上学期获 ICPC 亚洲区域赛上海站银奖，已免修数据结构课设，但他凭着对编程的热爱，仍全程参加这门课，并投入了巨大的精力，甚至在课程结束后，仍在 Botzone 上优化改进程序。	两院前 9 名中计算机学院占据 7 席，但冠军再次被软件学院朱志放同学获得。

2020 级？主角就是你们，请书写你们的历史，明年朱老师会把你们的精彩故事讲给下届学弟学妹听。为不给大家太多压力，本届仅开展这一项全员参与的对抗赛。

九、诚信要求：

通过查重和答辩确定是否抄袭或雇佣他人完成。违纪者视情节轻重倒扣分，完全抄袭他人或找他人代做的，本课程记为不及格。查重时样例代码和注释不算在内。答辩不用准备 ppt，只需讲解程序并回答老师提问即可。

本题有一个大坑，就是网上有可参考的代码。网上的代码可以参考，但不可直接照搬，将其他同学或网络上的代码改头换面，修改变量名或函数名、变换语句结构或函数位置等，均视为完全照搬、完全抄袭。

上一届计算机学院有同学完全照搬网上代码，被查重软件查出，该课程直接被记为不及格，目前正在和你们这届一起重修。上一届软件学院有多位同学，虽未 100% 照搬网上代码，但与网上代码雷同度过高，所以该题被判 0 分，其中不乏学习成绩优秀的同学。

老师已经下载了本题在 GitHub、CSDN、博客等网络上的所有免费和收费代码，而上届所有同学的代码也保存在老师家的电脑里，这些代码都会作为查重模板，你们的代码将会和上述代码一起查重。队内任何一名同学抄袭，则该组所有队员本课程均记为不及格！抄袭队与被抄袭队双方同论，不做区分，无论是否同班、同专业、同学院、同年级。因为是用软件查重，查 1000 份代码和查 2000 份代码，对于老师来说，成本是一样的。

我们允许并鼓励大家查阅资料、文献、学术论文，甚至自学一些高级技术完成本题。

任何时候若发现有违纪情况可发邮件到 jludatastruct@163.com 进行举报。

九、本题难么？

本题开放性很强，没有完全标准的答案，同学们可以充分发挥聪明才智和想象力，给出自己的解法。任何基础、任何层次的学生都有能力给出解决方案。

比如可以通过不断的玩游戏，找出一些经验技巧，进而构思一些规则，按照固定的规则决策。亦或是可以考察在当前局面下所有可能的下法中，执行哪种下法后产生的新局面对本方最有利（可以设定一个估值函数对每个局面进行评估，算一个分值），然后选取对本方最有利的下法。也可以多考虑几个回合，例如采用博弈树极大极小搜索或蒙特卡洛树搜索。亦或将上述几种方法相结合。

不鼓励采用机器学习方法训练神经网络，这不属于本课程的范畴，不允许使用 Python 或各类机器学习开源框架。如果你训练了神经网络，答辩时需将涉及的模型原理和细节都讲解清楚，否则视为抄袭。

希望大家不要随波逐流，不要看别人怎么做自己就怎么做，我更希望大家有自己的见解。

对于 Botzone 的使用、游戏规则与编程的问题除了问本班的老师外，还可直接在 Botzone 讨论区（NOGO 游戏）发帖，会有更为专业的北大 Botzone 开发团队人员回答。

对于本题，普通同学和竞赛大佬的差距可能会缩小。如果是 OJ 题目，竞赛选手经历了长期的训练，做题数量是普通同学的 10~20 倍以上，有着极为丰富的经验。但对于这种编程对战的形式，相信不少竞赛同学也是头一回经历。从去年的情况看，比赛中不乏普通同学战胜竞赛大佬的情况出现。所以对于普通同学来说，要有信心，只要努力，你也有机会。对于竞赛的同学来说，也不要轻敌，不要觉得简单弄弄就能夺冠，若想取得好成绩，还是需要精心投入。

本题也是北京大学 2020 级学生大一上学期《计算概论》（相当于我们的程序设计基础）的课程大作业，而我们已经是大二下学期。从去年的情况看，我们表现优秀的队伍即便和北大的前几名竞争，也是极具竞争力的。所以我们要充足自信。

当然，只要是比赛，就难免有不确定性和偶然性，比赛结果并不重要，这也是我将比赛结果所占比重仅设为 30% 的原因。与比赛结果相比，在这期间编程能力得到的锻炼和学到的新知识才是收获！