

欢迎加入2018年小学期实践教学——五子棋大作战

<div class='text-center'> </div>

2016年1月28日，DeepMind在《自然》杂志上公开他们的论文《Mastering the game of Go with deep neural networks and tree search》，AlphaGo成为了第一个可以战胜职业棋手的围棋AI，人工智能再次攻下了一个人类引以为傲的智商高地。2016年3月，AlphaGo又以4:1的成绩击败顶尖职业棋手李世石。2017年5月，AlphaGo和世界第一棋手柯洁比试，取得3:0全胜的战绩，中国围棋协会授予AlphaGo职业围棋九段的称号。

AlphaGo的成功使得人工智能的研究再度火热。这个夏天，我们也来打造一颗聪明的大脑，看他能不能比人更强悍。开始我们夏天的大作战吧！战场就是围棋盘上纵横的小天地，考虑到熟悉围棋的人不多，我们换个规则，我们进行一场“五子连珠”大作战。

五子棋是一种两人对弈的纯策略型棋类游戏，是世界智力运动会竞技项目之一，双方分别使用黑白两色的棋子，下在棋盘直线与横线的交叉点上，先形成五子连线者获胜。它起源于中国上古时代的传统黑白棋，主要流行于华人和汉字文化圈的国家以及欧美一些地区，是世界上最古老的棋。它容易上手，老少皆宜，而且趣味横生，引人入胜；不仅能增强思维能力，提高智力，而且富含哲理，有助于修身养性。现在点击上面的“开战” (/join_game)试试身手吧。

你知道和你下棋的是谁吗？可能是你的同学，也可能是一个机器打造的大脑。为了防止冷场，我们雇佣了16位身经百战的机器棋手，这里 (/context/ai_player)有它们的名单，看看哪个是你的手下败将。

我们介绍其中几个：

- alphago (/accounts/alphago) 这是AlphaGo的五子棋版本，不过不是DeepMind出品的，能力稍弱一些。它的介绍在这里 (<https://github.com/splendor-kill/ml-five>)，还有源代码哦。
- Yixin (/accounts/Yixin) 弈心是第13届、14届、15届、16届、17届、18届、19届Gomocup冠军。2017年，弈心成为首个在公开比赛中战胜人类顶尖棋手的人工智能程序。它的作者是上海交大的本科生孙锴，他给他的AI大脑做了一个网站 (<https://www.aiexp.info/pages/yixin-cn.html>)。

那么，我们开动吧 (/context/step_01)！

第一步

服务器是交战的战场，我们的AI大脑想要参战，先得找到去战场的路。

任务 1

写程序从以下网址取得下一步的指示

```
http://192.168.0.103:8000/step_01
```

参考资料

- <https://stackoverflow.com/questions/1011339/how-do-you-make-a-http-request-with-c>
(<https://stackoverflow.com/questions/1011339/how-do-you-make-a-http-request-with-c>)
- <https://stackoverflow.com/questions/1485708/how-do-i-do-a-http-get-in-java>
(<https://stackoverflow.com/questions/1485708/how-do-i-do-a-http-get-in-java>)
- <https://stackoverflow.com/questions/2023893/python-3-get-http-page>
(<https://stackoverflow.com/questions/2023893/python-3-get-http-page>)

第二步

恭喜你到达第二步！

警告：如果你是直接使用浏览器获得本页地址的话，请你返回第一步 (/context/step_01)。

在前一步里，你得到的这样的返回结果

```
{"is_success": true, "message": "Please visit http://192.168.0.103:8000/context/86e0caa3527526e88c3300ff4c2c3d6e"}
```

这是一个JSON，它是JavaScript Object Notation，JS对象标记。它采用完全独立于编程语言的文本格式来存储和表示数据，易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。它是计算机网络中最常用的数据交换语言。访问这个网站 (<http://www.runoob.com/json/json-tutorial.html>)了解更多内容。

在和服务器交流时，服务器总会返回一个JSON，其中 `is_success` 字段总是存在，它表示你的需求是否被满足，如果请求成功，`is_success=true`，否则 `is_success=false` 表示失败，此时还会返回一个 `error` 字段来说明出错的原因。

现在你已经能够和服务器取得联系了，就向服务器介绍一下你自己吧。

任务 2

请你告诉服务器你的名字是什么，你的学号是什么。名字使用字段 `name`，学号使用字段 `student_number`。请把以上信息以GET方式交到以下地址：

```
http://192.168.0.103:8000/step_02
```

提示：对你来说处理中文可能有点困难，你可以给自己起个英文名字。

第三步

恭喜你到达第三步！

你刚向计算机表明你是谁，但是怎么证明你就是你，而不是别人冒充的呢？

慢一点，这个问题并不是那么容易。为了证明你的身份，你需要向我们展示什么东西是只你有，而别人没有的。比如，身体特征，我们有人脸识别、指纹识别、虹膜识别；比如，证明文件，刚刚结束的考试上，你需要提供学生证、身份证；再比如，专属物品，你的钥匙、手机、银行卡、将军的虎符；还比如，你的秘密，密码、口令、咒语。

在网络上用真实物品来证明的办法不好用，你只有悄悄告诉服务器你的密码，不过互联网就是一个菜市场，你说的每一句话都会被有心人听到，所以得想个办法和服务器的悄悄话。我想这个技术是互联网发明以来最重要的东西了。去看看是什么办法能让你在菜市场和服务器的接上头。

任务 3

阅读以下材料，找出提示

链接: https://pan.baidu.com/s/1uNM4H_x63TPR2SHucfLIxw 密码: 3ps9

第四步

恭喜你到达第四步！

看完之后，你大概了解了公钥体系。我们来看个例子：

公钥是：

$p = (5, 493)$

私钥是：

$s = (269, 493)$

在计算机的世界里所有的信息都是以数字的形式存在的，我的秘密是一个数字 327，我要使用上面的密钥，利用公钥体系把它悄悄告诉服务器。

用公钥加密：

$message_s = 327^{5\%493} = 259$

这时发送的密信是 259，服务器收到这个密信，用私钥把它解密：

$message = 259^{269\%493} = 327$

服务器就得到了我发送的密码 327。

如你所见，这里要算很大数的幂，比如

$259^{269} =$
150534671041806386090299252242817157549079347221170002427772586345515804701401534122206804845
197908053816556137243966767051738575974030881841586258579793151358447737893795556235583025216
239970258912001385112746256684305193715371688999701811899745544537633720476267599699673950979
678605578193560302775196257694007707635860280571799376755700375869397597700830933538412528561
617685008593734568781033681677589068302772340184678271677797118904015176305335729216573901394
110439640352914561613130735767243277959601835324736713145668845297286826818230592751411388820
49683036832736391286971240023524693755814249666381072016621932311414629374478355180074544339

当然我们不需要知道它的结果是什么，我们只需要知道取模后的结果。这种运算在刚刚阅读的材料中叫做“钟算”，更普遍的叫法是“模幂”（modular exponentiation）

(<https://www.khanacademy.org/computing/computer->

science/cryptography/modarithmetic/a/modular-exponentiation), 有一个很快的算法。

任务 4

实现模幂算法, 通过服务器的检验。

访问 http://192.168.0.103:8000/step_04 服务器会给你10个问题, 每个问题包含三个数 (a, b, c) , 请给出 $a^b \pmod c$ 的值。返回值写入字段 `ans`, 10个数字用逗号, 隔开, 提交到 http://192.168.0.103:8000/step_04。

提示: 注意逗号必须是英文逗号。

第五步

恭喜你到达第五步!

我想你一定很艰难, 前一步的问题需要大数运算, 因为这个算法依赖于质因数分解的复杂度, 只有数字相当大时才能保证这个算法难于破解。

这是服务器使用的公钥:

```
65537,  
135261828916791946705313569652794581721330948863485438876915508683244111694485850733278569559  
191167660149469895899348939039437830613284874764820878002628686548956779897196112828969255650  
312573935871059275664474562666268163936821302832645284397530568872432109324825205567091066297  
960733513602409443790146687029
```

你需要把你的密码使用公钥加密后交给服务器, 当然在真正这么做之前, 我们先做些准备:

1. 验证: 将数字 31415926 使用公钥加密后作为 `num` 字段提交到
http://192.168.0.103:8000/step_05

如果没有错误, 我们进行下一步。我们这里加密的是数字, 而密码是个字符串, 因此需要把字符串转换为数字。转换规则是这样的: 把所有字符转换为ASCII, 之后把字符串看作一个256进制的数, 比如:

```
'abc'  
'a' = 97, 'b' = 98, 'c' = 99  
'abc' = 97*256^2 + 98*256^1 + 99*256^0 = 6382179
```

2. 验证: 将字符串 `hello, world!` 转换为数字后作为 `str2num` 字段提交到
http://192.168.0.103:8000/step_05

如果没有错误, 我们进行下一步。转换后的字符串可以加密了。

3. 验证: 将字符串 `hello, world!` 转换为数字, 使用公钥加密后作为 `str` 字段提交到
http://192.168.0.103:8000/step_05

如果没有错误, 我们进行下一步。加密后的数字可以用16进制表示, 这样会短一点, 我们秘钥长度为1024字节, 使用16进制, 一位16进制数可以表示4个字节, 生成的加密数据有256位长, 可以节省一点网络流量。

4. 验证：将字符串 `hello, world!` 转换为数字，使用公钥加密后转换为16进制，作为 `hex` 字段提交到 `http://192.168.0.103:8000/step_05`

如果没有错误，我们进行下一步。你可以把你的密码加密了。

任务 5

把你的用户名作为 `user` 字段，加密后的密码作为 `password` 字段，提交到 `http://192.168.0.103:8000/step_05`

不要急着找下一步的地址。作为一个确定性的程序，你的密码每次加密得到的结果都是一样的，我们可以为密码加噪，扰乱窃密者。在加密时，我们可以编码的信息不大于公钥中的第二个数字，这个数字很大，因此我们可以编码很长的信息。

在c/c++中字符串以字符 `\0` 作为结束标志，`\0` 以后的所有数据都会被忽略掉，因此我们可以在密码后面添加一个字符 `\0`，之后连接上一个随机生成的没用的字符串。这样每次都会产生不同的密文，就能更有效的保护你的密码了。去试试吧！

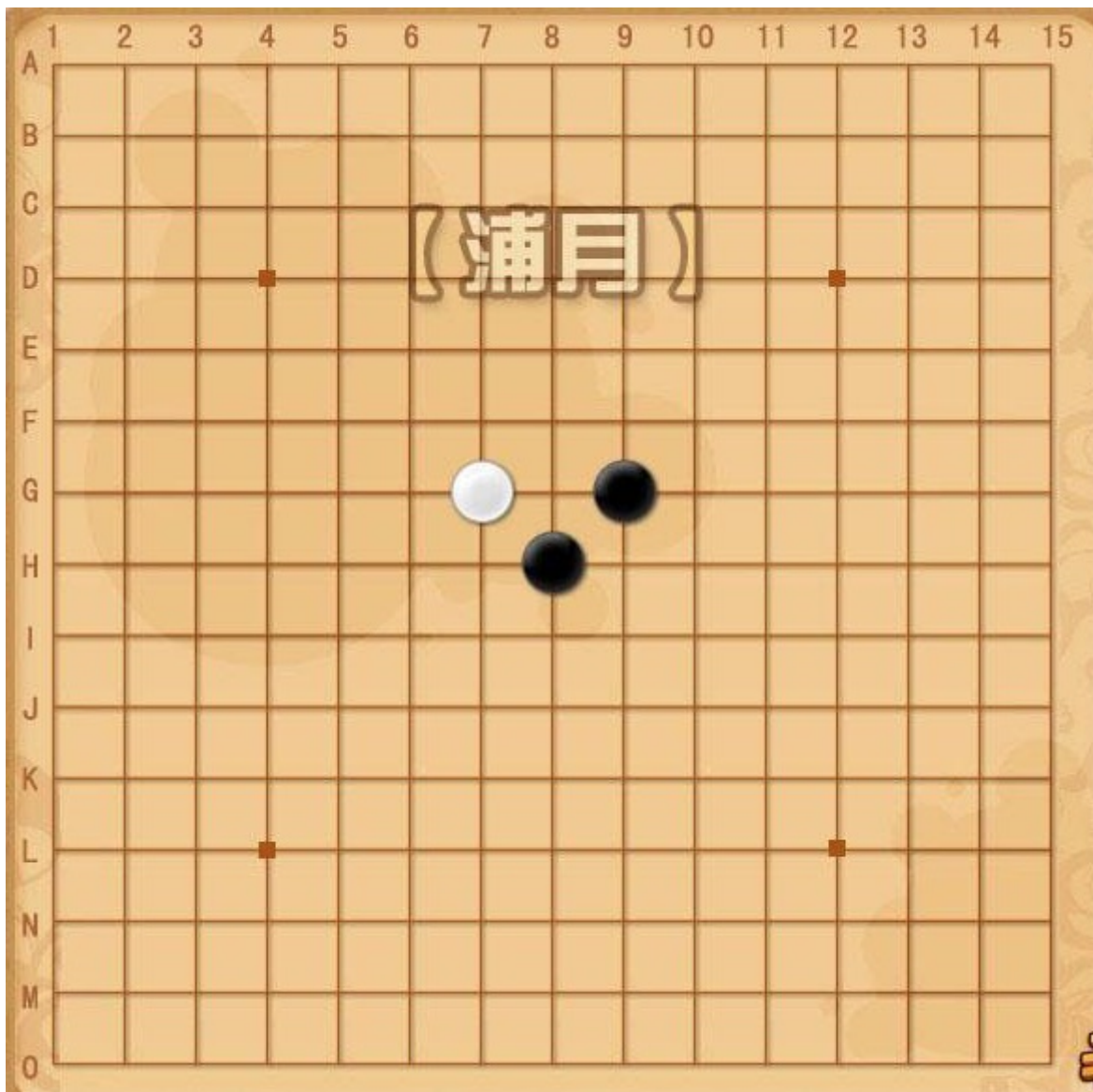
回到我们得到的信息，它也是被加密的，你需要解密，服务器使用它自己的私钥加密了数据，因为秘钥是对称的，你可以用公钥解密它。

第六步

恭喜你到达第六步！

你已经成功实现了公钥体系最为关键的部分。现在服务器相信你就是你了，下面开始你的战斗。

五子棋的棋盘有15×15个交点，一共有225个交点，我们可以在每一个交点上放置棋子。方便起见，我们为每一个交点起一个名字。



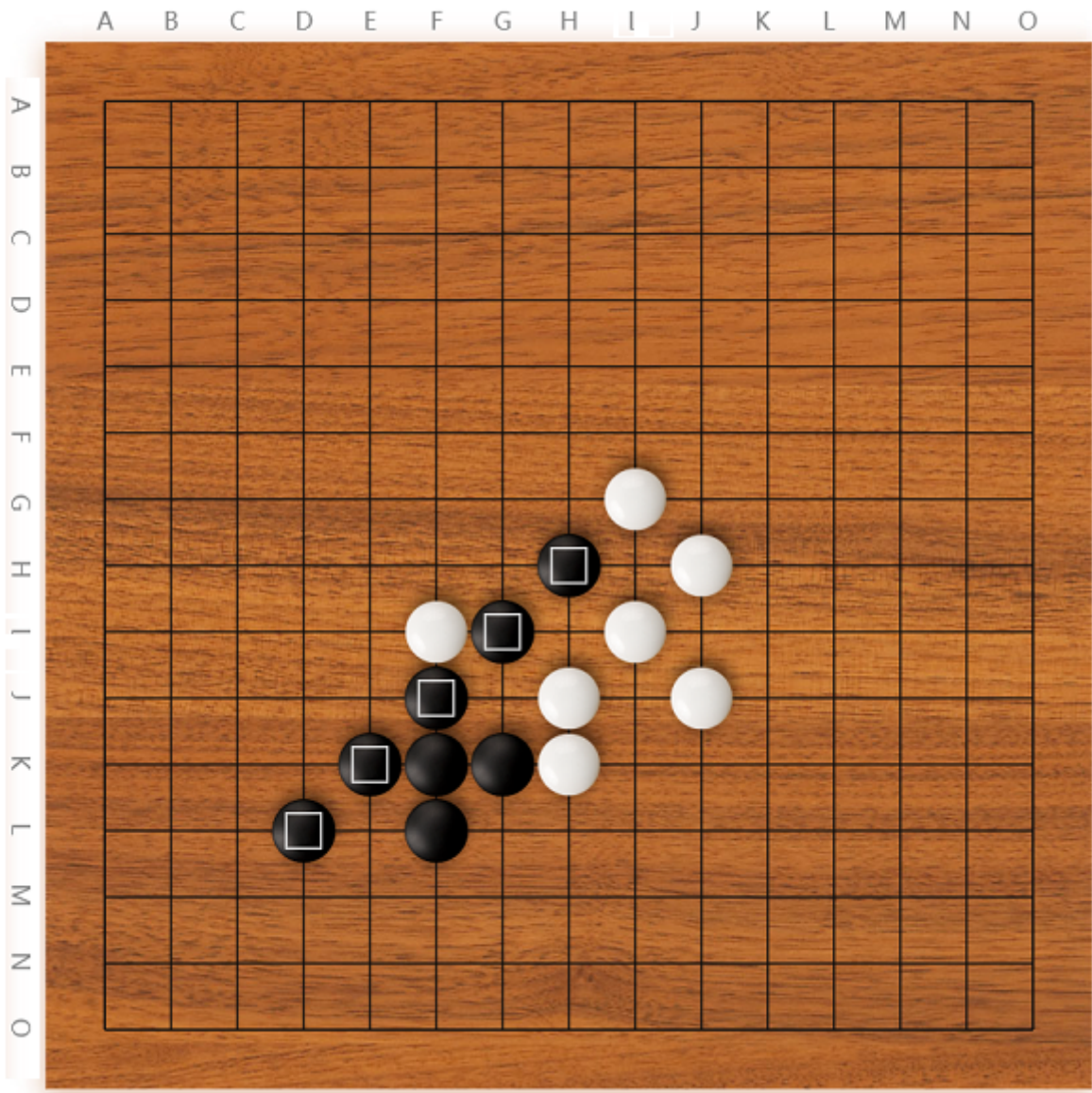
在上面的图上，我们为每一行标上一个字母从A到O，每一列标上一个数字，从1到15。我们用行和列的标号表示一个交点，比如图上的白棋的坐标是 G7。

这种编码策略有一个问题，它是变长的，比如左上角的交点的坐标是 A1 有2个字符，而右上角的交点的坐标是 A15 有3个字符。我们把每一列也标上一个字母从A到O，这样不管是哪个交点的坐标都是两个字符了。左上角的交点的坐标是 AA、右上角的交点的坐标是 AO、那个白棋的坐标是 GG。

游戏开始前，棋盘是空的，根据规则，黑方先落子，每落一子我们就记录一下它的坐标，最后就会得到类似这样的字符串

```
HHJHKGII LFHJJFJJKFIFKEKHIGGILD
```

它对应的棋局是这样的：



你可以逐步跟踪它的落子顺序。
注意这里的标号和游戏中是不同的。给考生增加困扰是所有出题人的本意，才不承认是犯了错



呢！
为了保证你和服务器使用相同的棋盘约定，我们把上面的坐标形式转换为图形形式，我们用 . 表示空白位置，x 表示黑棋，o 表示白棋。那么上面的坐标表示就是这样的：

```
.....
.....
.....
.....
.....
.....
.....
.....O.....
.....X.O.....
.....OX.O.....
.....X.O.O.....
.....XXXO.....
...X.X.....
.....
.....
.....
```

任务 6

实现从坐标表示到图形表示的算法，通过服务器的检验。

访问 http://192.168.0.103:8000/step_06 服务器会给你一个坐标表示，请给出每一步的棋盘图形表示，如上面的例子共有15步，其中第1步：

```
.....
.....
.....
.....
.....
.....
.....
.....X.....
.....
.....
.....
.....
.....
.....
.....
```

第2步：

.....

.....

.....

.....

.....

.....

.....

.....

.....X.....

.....

.....O.....

.....

.....

.....

.....

.....

第3步：

.....

.....

.....

.....

.....

.....

.....

.....

.....X.....

.....

.....O.....

.....

.....X.....

.....

.....

.....

第4步：

.....

.....

.....

.....

.....

.....

.....

.....

.....X.....

.....O.....

.....O.....

.....X.....

.....

.....

.....

.....

第5步：

.....
.....
.....
.....
.....
.....
.....
.....
.....x.....
.....o.....
.....o.....
.....x.....
.....x.....
.....
.....
.....

等等等等。

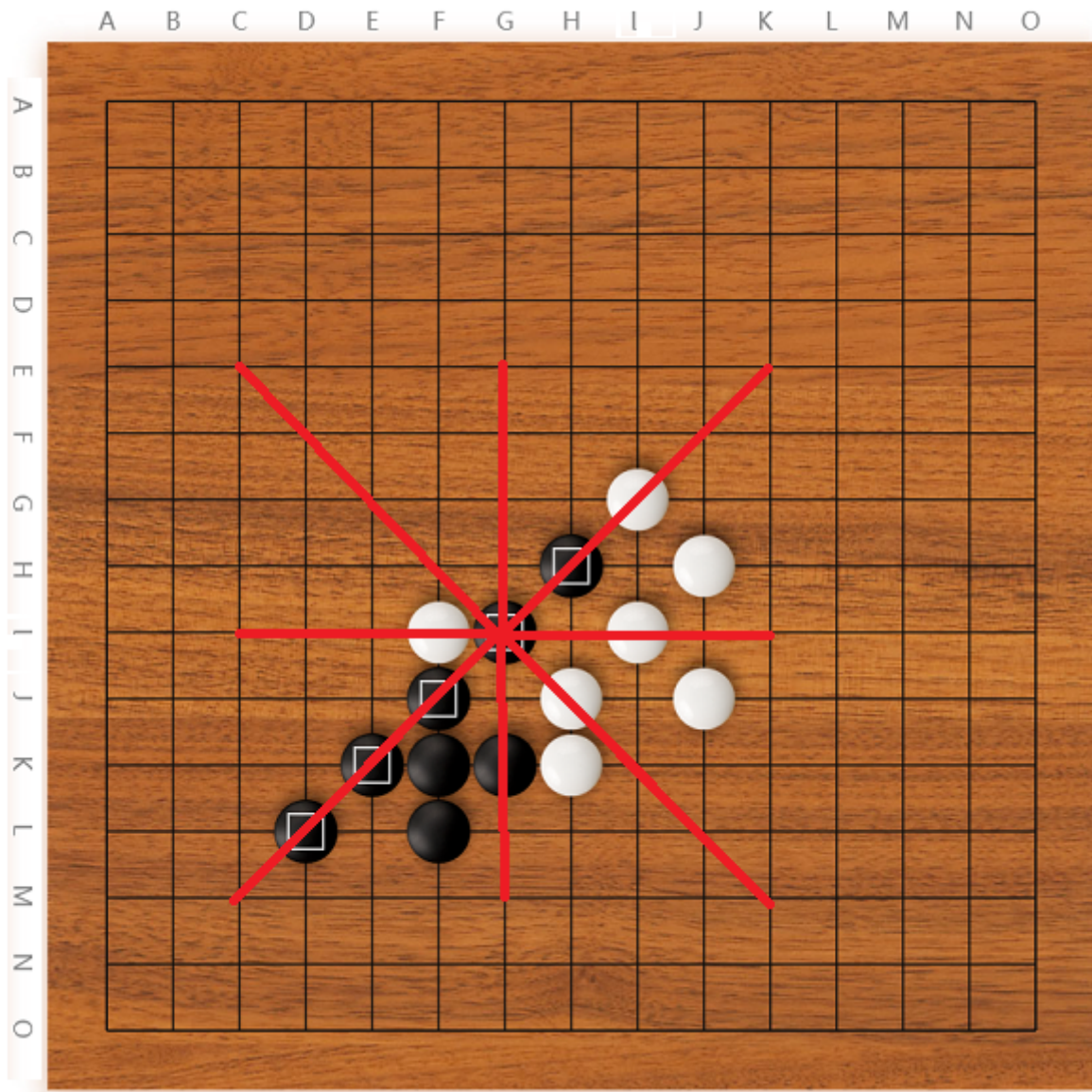
请将每一步的棋盘图形表示，按顺序写入字段 ans，每步用逗号，隔开，图形表示中换行可省略。结果提交到 http://192.168.0.103:8000/step_06。

第七步

恭喜你到达第七步！

到这一步，你对战场环境就有了充分的了解。下面我们准备打造机器大脑了。你可能已经有你自己的想法了，不过别着急，先跟着我的思路做。我会提示你解决一些基本的问题。

棋局的分析关键在于分析棋型，在五子棋中，我们仅需要考虑一条线上的棋子分布，一个点同时属于4条线，如下图红线所示，其中交点为要考察的点：



由于五子棋仅需要考虑五个棋子成为一条线的情况，我们只需要考虑要考察的点周围5个格子范围内的点。

想一想



任务 7

实现模型算法，通过服务器的检验。

访问 http://192.168.0.103:8000/step_07 服务器会给你一个棋盘的坐标表示（board 字段），以及一系列要考察的点的坐标（coord 字段），坐标之间用逗号，隔开，整体用 [] 包围，事实上它是一个JSON的数组 (<http://www.runoob.com/json/json-syntax.html>)。

- 请给出每个考察点的四条线上的棋型
- 只考虑周围5个格子的范围
- 用 . 表示空白位置，x 表示黑棋，o 表示白棋。

- 每个考察点返回四个表示棋型的字符串
- 所有考察点的棋型字符串按顺序返回，其中单个考察点的棋型字符串顺序、棋型字符串中棋子的顺序不做规定
- 返回值写入 ans 字段，提交到 http://192.168.0.103:8000/step_07

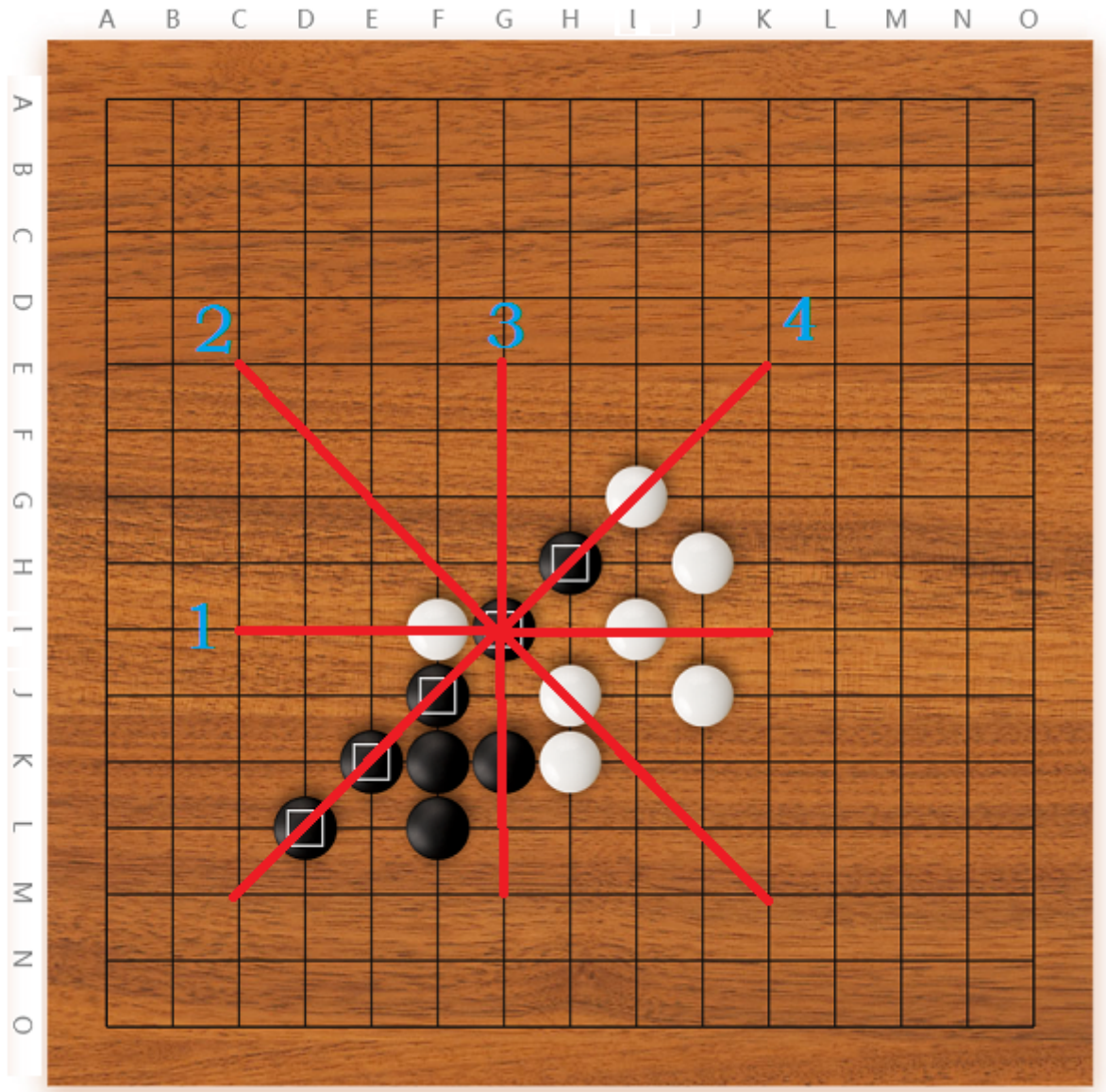
如上例中，问题为：

```
{'board': 'HHJHKGIILFHJJFJJKFIFKEKHIGGILD', 'coord': ['IG']}
```

考察点 IG 的四条线上的棋型分别为

```
...OX.O..  
....XO...  
....X.X..  
..OXXXXX.
```

上面的棋型按照下图中标出的顺序给出，不过你可以自己选择你觉得方便的顺序。



我们最后把所有的棋型字符串用逗号隔开，依次写入 ans 字段。

第八步

恭喜你到达第八步！

利用前一步得到的棋型分析结果，考察每一个可能落子的位置，给每一个可能的位置打分，将棋子落在分数最高的位置上。根据经验，我们可以总结出下面的落子规则：

1. 致胜棋型

若在某处落子后我方获胜，采取这一落子位置。

我们将空位子记作 . ，本方棋子记作 M (me) ，对方棋子记作 o (opponent) ，考察点记作 c (current) ，能够致胜的棋型必然包含：

```
"CMMMM"  
"MCMMM"  
"MMCMM"  
"MMMCM"  
"MMMMC"
```

为了保证其优先级，我们给这条规则分配一个大的分数 10000 。

2. 必须防守棋型

若我方棋子不落在某处，对方将出现致胜棋型，采取这一落子位置，不过优先级小于前者。

按照上面的记号，此时棋型应包含：

```
"O000C"  
"C000O"
```

我们给它分配一个较大的分数 6000

3. 两步致胜棋型

若在某处落子后，出现致胜棋型，采取这一落子位置，不过优先级小于前者。

按照上面的记号，此时棋型应包含：

```
".CMMM."  
".MCMM."  
".MMCM."  
".MMMC."
```

我们给它分配一个较大的分数 5000

4. 两步防守棋型

若我方棋子不落在某处，对方将出现两步致胜棋型，采取这一落子位置，不过优先级小于前者。

```
"C000."  
".000C"  
".00C0."  
".0C00."
```

我们给它分配分数 2500

5. 进攻模型

若在某处落子后，出现一端被挡住的四个连续棋子，此时对方必需挡住另一端，采取这一落子位置，不过优先级小于前者。

按照上面的记号，此时模型应包含：

```
"OCMM."
"OMCMM."
"OMMCM."
"OMMMC."
".CMMMO"
".MCMMO"
".MMCMO"
".MMMCO"
```

我们给它分配分数 2000

6. 两步进攻模型

若在某处落子后，出现两端都不被挡住的三个连续棋子，此时对方必需挡住其中一端，采取这一落子位置，不过优先级小于前者。

```
".MMC."
".MCM."
".CMM."
```

我们给它分配分数 400

7. 预防模型

若我方棋子不落在某处，对方将出现两步进攻模型或进攻模型，采取这一落子位置。

```
".OOC"
"COO."
"MOOOC"
"COOOM"
```

我们给它分配分数 400

8. 无效进攻防守模型

若在某处落子后，出现一端被挡住的三个连续棋子，或者若我方棋子不落在某处，对方将出现一端被挡住的三个连续棋子，这一步进攻或者防守不会引起对方的立即反应。

```
".MMCO"
".MCMO"
".CMMO"
"OMMC."
"OMCM."
"OCMM."
"MOOC"
"COOM"
```

我们给它分配分数 200

9. 布局棋型

若在某处落子后，出现两端都不被挡住的两个连续棋子。

```
".MC."  
".CM."
```

我们给它分配分数 50

10. 其他棋型

我们给它分配分数 20

再次说明，我们的方法并不是最好，但请你先实现这种方法，之后再尝试你的想法。

我们将计算每一个当前还是空着的位置的得分。

1. 按照第六步的方法获得棋盘，此时需要根据当前落子是黑方还是白方，将黑棋和白棋表示为 M 或 O
2. 找到所有还空着的位置
3. 对每一个空位子
 - a. 将这个位置设为 c
 - b. 按照第七步的方法获得该位置的棋型字符串
 - c. 在所有的棋型字符串上按顺序寻找上述所有模式，每得到一个匹配，将对应的分数加上，最终所得即为该位置的分数
 - d. 将这个位置设为 .
4. 找出分数最大的位置，若分数最大的位置有多个，随意选择一个

任务 8

实现按照评分规则选择落子位置的算法，通过服务器的检验。

访问 http://192.168.0.103:8000/step_08 服务器会给你几个棋局的坐标表示，保存到 questions 字段，以JSON数组表示。请给出分数最大的一个落子位置，按顺序给出坐标，坐标之间用逗号隔开。服务器不会告诉你，你是白棋还是黑棋，你需要根据规则自己确定。

第九步

恭喜你到达第九步！

上一步我们已经完成了一个AI大脑的最核心功能。我们可以用它来对战了。

访问服务器 http://192.168.0.103:8000/join_game，会返回一个游戏编号 game_id。之后你可以使用这个游戏编号，进行游戏 http://192.168.0.103:8000/play_game/{game_id} 并查询游戏状态 http://192.168.0.103:8000/check_game/{game_id}。

利用这三个功能我们就可以让我们的AI参战了。这个过程应该是这样的，这是一个典型的消息循环。

1. 用 `join_game` 加入游戏
2. 用 `check_game` 检查游戏状态
 - a. 如果游戏完成就退出
 - b. 如果不轮你下，就等一会，否则使用AI确定要落子的位置，并用 `play_game` 告知服务器你落子的位置
 - c. 返回到第2步用 `check_game` 检查游戏状态

其中 `join_game` 需要登录，需要提交用户名和密码，需要使用第五步使用的加密方法对密码加密，用户名写入 `user` 字段，加密后的密码写入 `password` 字段。返回字段 `game_id` 是加入的游戏编号。

`play_game` 也需要登录，落子的坐标写入 `coord` 字段。

`check_game` 不需要登录，返回当前的游戏状态，会返回以下状态：

- `is_success` , `error` 查询是否成功及错误原因
- `step` 当前是第几步
- `creator` 游戏一方的用户名
- `creator_name` 游戏一方的名字（昵称）
- `creator_stone` 游戏一方使用的棋子（x 表示黑棋，o 表示白棋）
- `opponent` 游戏另一方的用户名
- `opponent_name` 游戏另一方的名字（昵称）
- `opponent_stone` 游戏另一方使用的棋子（x 表示黑棋，o 表示白棋）
- `ready` 游戏是否就绪，两个玩家都在线时，游戏进入就绪状态
- `current_turn` 当前应当落子的玩家的用户名
- `current_stone` 当前应当落子的玩家的使用的棋子（x 表示黑棋，o 表示白棋）
- `left_time` 剩余时间，为避免玩家过长思考，限制玩家必须在60秒内落子，否则游戏结束
- `winner` 获胜的玩家的用户名，当游戏没有产生赢家时，该值为 `None`
- `board` 棋盘的坐标表示
- `last_step` 上一步落子的坐标
- `win_step` 如果一方获胜，这个字段给出连成五子的一条线的棋子坐标

注意不要过于频繁的检查游戏的状态，使用 `sleep` 函数等待服务器更新状态，两次检查以5到10秒的间隔为宜。。

任务 9

实现消息循环，开始作战吧！

下一步 (/context/8220445bdd18916cefd7f651073168)

第十步

恭喜你到达第十步！

你已经完成了一个AI的设计，下面就需要发动你的智慧让你的机器大脑变得更聪明了！

我们的征途是星辰大海!

更多资料可以参考弈心作者给出的列表 (<https://www.aiexp.info/gomoku-renju-resources-an-overview.html>)。

课程成绩将根据你的AI大脑的对战成绩决定，我们采用Elo评分体系 (https://en.wikipedia.org/wiki/Elo_rating_system)。每次对战的胜负都会体现到分数上面，出发吧！

