Enigma-Cracker

以已知明文攻击的方式破解 Enigma 机的密钥。

参考资料：

- 知乎关于 Enigma 机及其破解的讲解：https://www.zhihu.com/question/28397034
- Enigma 机模拟器：https://www.101computing.net/enigma-machine-emulator
- 关于 Bombe 机的详细讲解：https://www.mpoweruk.com/enigma.htm
- Wiki 上的 Enigma 机：https://en.wikipedia.org/wiki/Enigma_machine
- Wiki 上的 Enigma 机破解分析：https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma

程序设计

运行环境

- 运行 LoopAnalyzer 需要在系统和 python 中均安装 Graphviz，并且将 Graphviz 放入 PATH 系统变量。Graphviz：http://www.graphviz.org/download/

程序功能

- 实现了 Enigma 类用于模拟 Enigma 机，支持五种转子、插线板、ring setting、初始位置的设置，完整按照 Enigma 机实现（包括有 double-stepping 特性）；
- 实现了 LoopAnalyzer 类用于辅助 loop 分析；
- 实现了 Bombe 类参照 Bombe 机算法进行破解，注意这里指定了要破解的 Enigma 机的 ring setting；
- 在破解的结果中，**除了得到转子类型和顺序，还会尽可能还原 plug_board**，如果有一些未还原的插线板配置，则可以很容易通过词汇比较的方式得到还原。

程序使用

- 【额外功能：在 enigma.py 中修改 `__main__` 下的代码，运行 `python enigma.py` 即可模拟 Enigma 机加密过程，可用于辅助分析】；
- 设置 input.json，运行 `python loop_analyzer.py` 即可辅助进行环分析，input.json 必要内容如下

```
{
  "plaintext": "ABBC",
  "ciphertext": "DCCA"
}
```

- 设置 input.json，运行 `python bombe.py` 即可模拟 bombe 机运行，input.json 格式如下：

```
{
  "ring_setting": "FEN",
  "offset": 5,
  "plaintext": "TSINGHUAUNIVERSITY",
  "ciphertext": "UUXQHFTSVFDUTXOYQV",
  "central_letter": "U"
}
```

其中 central_letter 即寻找到的所在环最多的结点。

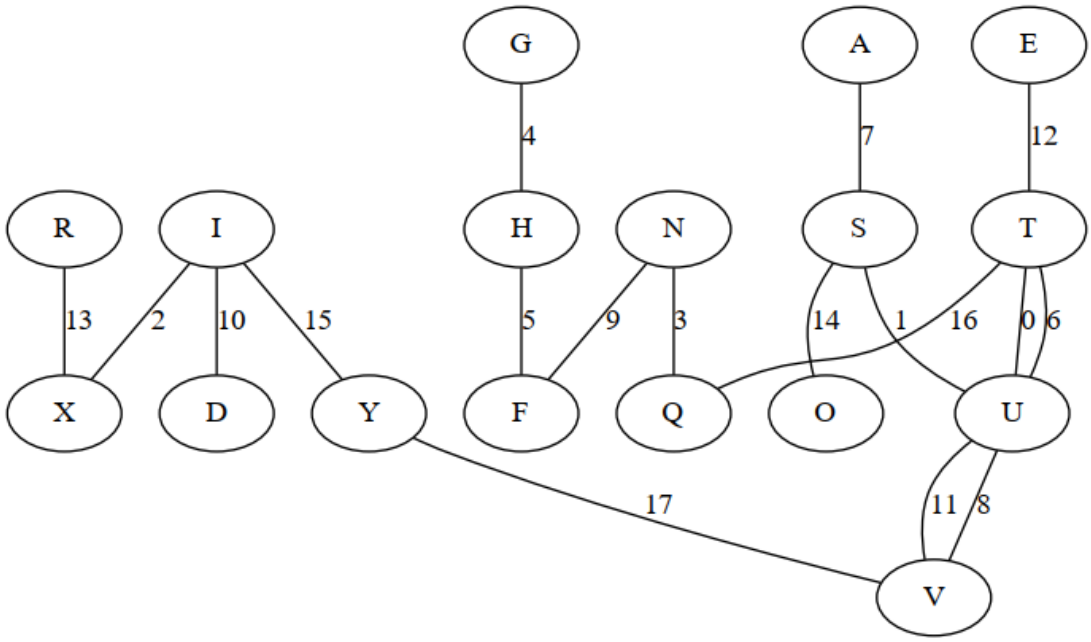
运行实例

假设加密文档为：`UZDTIUUXQHFTSVFDUTXOYQV`，在文档第六个字母处找到了一个 crib，对应明文为 "TSINGHUAUNIVERSITY"，密文为 "UUXQHFTSVFDUTXOYQV"

若已知 ring_setting 为 "FEN"，则设定 input.json 如下：

```
{
  "offset": 5,
  "ring_setting": "FEN",
  "plaintext": "TSINGHUAUNIVERSITY",
  "ciphertext": "UUXQHFTSVFDUTXOYQV"
}
```

运行 `python loop_analyzer.py` 得到图如下（此时暂未使用 offset 和 ring_setting）：



发现字母 U 所在环最多（有两个），所以更新 input.json 如下：

```
{
  "offset": 5,
  "ring_setting": "FEN",
  "plaintext": "TSINGHUAUNIVERSITY",
  "ciphertext": "UUXQHFTSVFDUTXOYQV",
  "central_letter": "U"
}
```

运行 `python bombe.py` 进行运算，大约 18 秒运行一个设置，也就是 18 分钟之内能够枚举完所有 60 个设置。在运行到 [3, 4, 1] 这个转子顺序时，程序输出了：

```
INFO - {'rotors': [3, 4, 1], 'ring_setting': 'FEN', 'position': 'SCM',
'plug_board': {'U': 'U', 'F': 'T', 'T': 'F', 'E': 'V', 'V': 'E', 'N': 'S', 'S':
'N', 'Q': 'Q', 'D': 'H', 'H': 'D', 'Y': 'Y', 'O': 'O', 'B': 'A', 'A': 'B', 'P':
'G', 'G': 'P', 'M': 'I', 'I': 'M', 'X': 'X', 'Z': 'R', 'R': 'Z'}}
```

表示一个可能的设置，按照该设置，对原文档 `UZDTIUUXQHFTSVFDUTXOYQV` 还原出的文档为：`ILOVETSINGHUAUNIVERSITY`，解密成功。并且实际上程序只找到了这一个可能的设置。

Enigma 机

Enigma 机设计

此处讨论的是在二战中的 Enigma 机，此时德军对转子进行了改造，使得转子芯外面的字母圈可以绕着转子旋转，并且一共有五个转子，可以任意选择其中三个以任意排列方式使用。

日密钥

而德军的日密钥（即每月的密码本上记录的每日密钥）分为了三部分：

- 从五个转子中选择三个特定的转子，并按一定顺序排列；
- 每个转子外侧的字母圈相对于转子芯的位置，即 ring setting；
- 插线板所交换的 10 对字母。

注意这里已经不存在每日通用的转子初始位置，而是每次由操作员自己选择转子初始位置和信息密钥。

转子的旋转

首先令转子从左到右依次从“高”到“低”，分别称为转子1、转子2、转子3。每次按键转子3一定会转动。

转动原理是：每个转子上都有带 26 齿的棘轮，并且每个转子的字母环会“遮住”其左边相邻转子的棘轮（每个转子的字母环上会有一处凹痕），而每次按键时三个棘爪会同时试图推动棘轮，只有当某棘轮没有被右边转子的字母环遮住时才会被推动。需要注意的是，成功推动某个转子的棘轮时，由于同时卡入了右边转子的凹痕，所以右边的转子也会被推动。

于是结果就是：

- 每次按键，转子 3 一定会转动到下一个位置；
- 当转子 3 的字母环凹痕与棘爪对齐时，按键会使转子 2 转到下一个位置（此时第一个转子也会由于凹痕对齐而被同时推动，但由于转子 1 的棘轮本身就被推动了，所以看起来无影响）；
- 当转子 2 的字母环凹痕与棘爪对齐时，按键会使转子 1 转到下一个位置，此时由于凹痕被推动，转子 2 也会转到下一个位置。

这时就有了一个特性 **double-stepping**：当按键使转子 2 转到下一个位置，且该位置使得转子 2 的字母环凹痕与棘爪对齐时，下次按键转子 2 仍会转动，即在转子 2 在两次连续按键中都转动了。

五种转子的凹痕（当转动到该字母时，凹痕将与棘爪对齐）：

```
I: Q
II: E
III: V
IV: J
V: Z
```

五种转子转芯的映射：

```
I: 'EKMFLGDQVZNTOWYHXUSPAIBRCJ'
II: 'AJDKSIRUXBLHWTMCQGZNPYFVOE'
III: 'BDFHJLCPRTXVZNYEIWGAKMUSQO'
IV: 'ESOVPZJAYQUIRHXLNFTGKDCMWB'
V: 'VZBRGITYUPSDNHLXAWMJQOFECK'
```

这里使用的 UKW-B 反射器：

```
'YRUHQSLDPXNGOKMIEBFZCWVJAT'
```

加密

操作员加密的步骤为：

- 根据日密钥设置转子顺序和转子外侧字母圈相对于转子芯的位置，以及插线板交换的字母对；
- 自行选择转子初始位置和信息密钥，假设分别为 ABC 和 XYZ；
- 明文发送 ABC，然后以 ABC 作为初始位置，发送加密后的 XYZXYZ；
- 将转子初始位置设置为 XYZ，加密信息正文并发送。

解密

另一方操作员解密的步骤为：

- 根据日密钥设置转子顺序和转子外侧字母圈相对于转子芯的位置，以及插线板交换的字母对；
- 接收到明文传输的转子初始位置 ABC，对转子初始位置进行设置后，解密接下来的六个字母得到 XYZXYZ；
- 将转子初始位置设置为 XYZ，解密接下来的传输内容得到信息正文。

Bombe 机

由 36*3 个转子，即 36 个模拟的 Enigma 机（称为 Scrambler）组成，分为了三排，每排 12 个 Scrambler。不同排可以运行不同的转子设置和顺序，而每一排可以通过接线设置多个 loop （即从 crib 中分析得到的 loop）。

在这里合法性判断分为了两个部分：loop 和插线板映射唯一性判断。为了方便描述，将明文和密文这些能够直接看到的称为“插线板外的字符“，经过一次插线板后称为“插线板内的字符”。

- Loop：即课程 PPT 上写的内容，将字母作为结点，对应的明密文字母连线，找到一个所在环最多的字母 A（称为中心字母），对它经过插线板后的字母 X 进行枚举 —— 注意 X 在依次经过环之后应当保持不变；
- 映射唯一性判断：Loop 枚举过程中，会对于确定的一个字母 A 枚举经过插线板后得到的字母 X，此时其实可以得到整个连通图的插线板内字符，也就能得到插线板的其它位置映射关系，此时判断一下映射唯一性即可。

事实上，这两部分可以直接通过一次 BFS 得以实现，这个 BFS 实际上就是当时电路所做的事情——在上面所述的“得到整个连通图的拆线板”时，顺便也就把判环是否合法处理了。

具体做法即：

- 枚举中心字母 A 经过插线板后的字符 X；
- 找到明密文中字母 A 的所有出现处，推导出其经过插线板后的字符均为 X，此时经过对应的 Enigma 机便可得到另外一对插线板对，假设经过 Enigma 机后为 Y，对应的明或密文为 B，此时判断唯一性以及是否与之前的推导冲突；
- 再找到明密文中字母 B 的所有出现处，重复上述步骤即可（实际上就是一个 BFS）。

所以推导时找环的唯一目的事实上只是判断该 crib 的优劣，并且找到合适的枚举起点。