

课堂体验

课堂上关于图灵机和停机问题的理解有，但不够，只有在调试模拟程序的过程中才最终加深了理解，明白了Instructions对于图灵机的意义，明白了图灵机对于数据的表示方式，我觉得如果要增加图灵机的程序可读性，需要给出一些注释：

- 关于这台图灵机所要实现的功能的叙述
 - 做加法？做乘法？做乘法之后再加法？
- 关于这台图灵机的输入输出格式约定
 - 用单个0分割数字？用连续的1表示数字？
 - 停机时探头的位置？
- 关于实现该功能的算法思路
 - 顶层逻辑：复制元素？搬运元素？
 - 细节操作：（例如乘2）用状态1充当搬运工，用状态2、3复制元素，读取到末尾0的时候折返，再去搬运新的一批元素。

学习Python

作为有过信竞基础，能够编写面向过程的C++程序的同学，面向对象的学习过程花费了我不少的时间。关键不是在调用方式，而是在理解类的行为逻辑上。同时，从信竞面向过程的编程思维带来的坏习惯就是对全局变量的过分依赖。离开了全局变量，连一个简单的深度优先搜索都很难实现。本来打算通过再写一个类去解决，但最终还是向时间妥协，采用了几个全局变量来实现算法，最终使得main.py看上去不那么精炼，有些杂乱。

不过在学习Python的字符串操作、list操作和文件读写操作的过程中，我的确感受到了Python对于数据处理的强大。它将所有看上去是数据类型的东西都写成了类，提供了许多非常实用而方便的函数，例如字符串的

```
'{}'.format()  
''.join()  
''.split()
```

这三者，极大地帮助我处理好了图灵机的Instructions。

只不过Python的效率确实不尽如人意，跑完2w个图灵机花去将近半分钟，当然这其中也有很多可以优化的内容，例如对于'1,1R1,1R1'这种可以直接判断出无限循环的样例可以直接剪枝节省时间，省下第二状态的遍历时间，应该会使得算法更加高效。

孙权恩2021080101013
2022.3.26