第三周实验报告

1. 实验概览

本实验旨在理解时间复杂度概念、学习哈希散列的原理与实现、学习布隆过滤器的数学原理，理解其实现和内部参数选择，并基于框架实现布隆过滤器；理解并发编程的概念和原理并基于threading库实现并发爬虫。

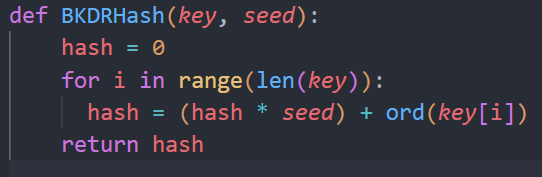
1. 实验环境

Docker: SJTU-EE208

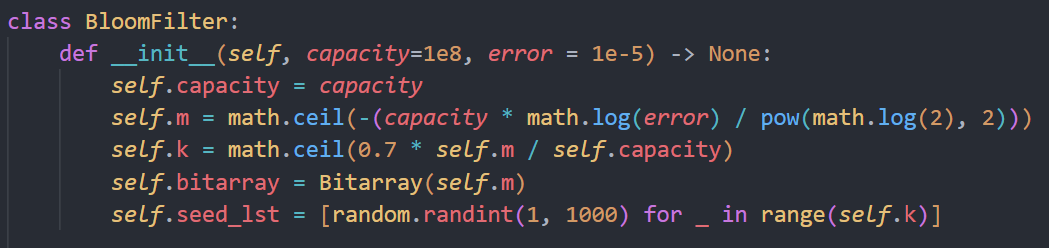
1. 解决思路
2. 练习一

修改提供的Bitarray.py，利用类实现布隆过滤器。

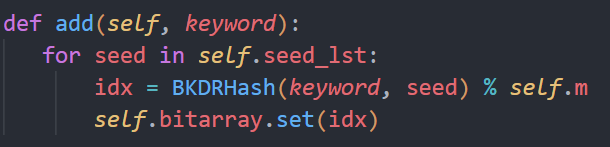
将GeneralHashFunctions.py文件中提供的BKDRHash函数修改为可传入seed的hash函数



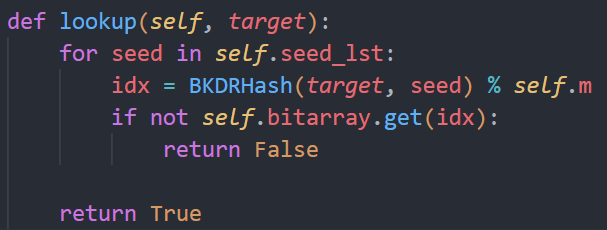
在初始化中定义过滤器中传入的字符串数量并根据参考文献得出对应的哈希函数个数和位数组的长度，初始化种子列表用于确定哈希函数



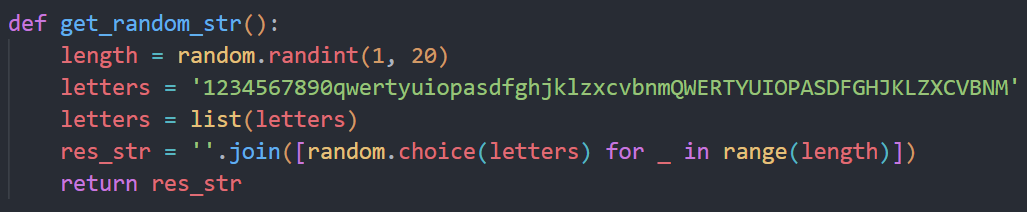
加入关键字：根据哈希函数个数得到每个关键字对应置1的位并修改位数组

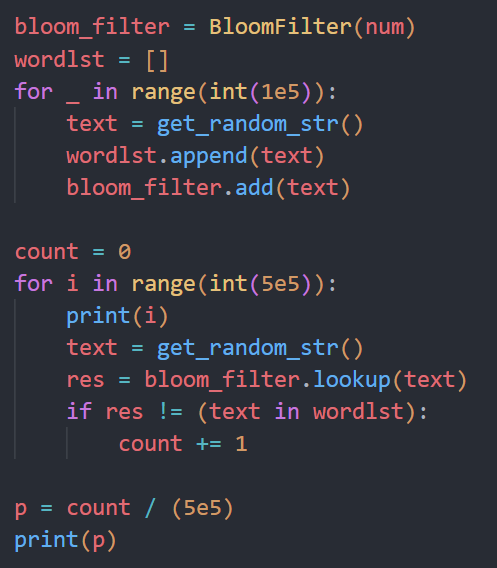


查找：利用上述哈希函数得到目标字符串的对应置1的位并检索位数组，如果对应位上的值均是1则判断该字符串在关键字中



测试：设计随机生成字符串的函数（测试中设置字符串长度为2-20位），并测试错误率



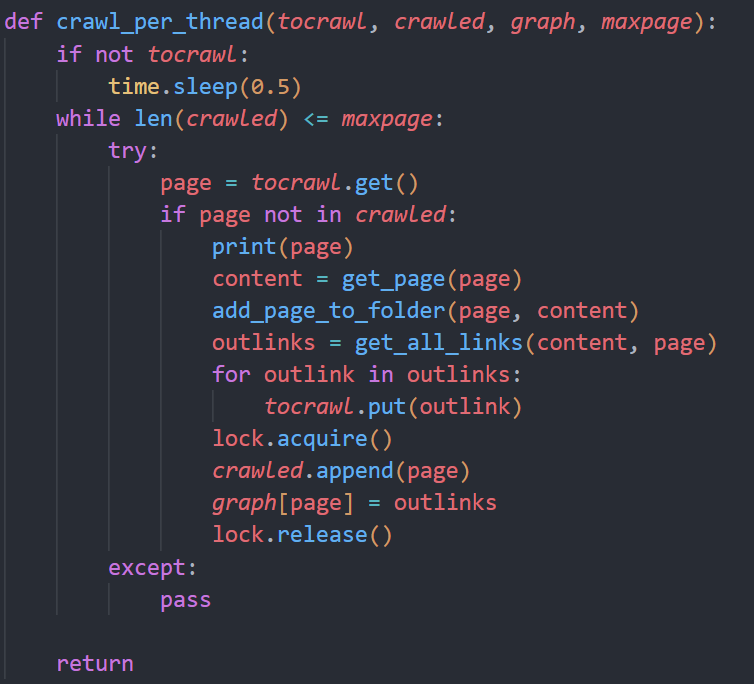


1. 练习二

修改lab2中的crawler.py文件，引入threading库，将工作分布给四个线程



利用threading.lock和queue处理并发程序中访问同一变量（queue自带并发处理，其中crawled和graph变量利用Lock类处理）的问题



1. 实验结果

练习一：经检验错误率约为7e-5，满足使用需求



练习二：结果详见index.txt 和文件夹html，下图为index.txt的部分截图



1. 分析与思考

线程和进程：进程是一个正在执行的程序的实例，包括程序计数器、寄存器和程序变量的当前值。进程之间是相互独立的，有自己的内存空间，而线程之间可以共享进程中的内存（因此存在访问同一变量的问题），线程具有的独立部分在于栈区和所用到的寄存器。多线程中抢占变量的问题在于处于临界区的线程存在地址冲突。