创新思维第六次作业感受

虽然说创新思维课只是一门开眼界、拓思维的小课，课时不长，只有半学期，但这门课在我心中一直是目前为止遇到过最硬核也最有收获的一门课。无论是课上内容的广度、深度，还是课下实践的无上限，都让我学到很多。
这次作业是最后一次，我依然感谢李文老师和顾实老师的课程设计、课上讲授和课下答疑，没有你们的工作，我们很难有机会在大一接触这么多有趣的计算机知识，差分机、分析机、图灵机，P=NP问题、数据安全、PageRank算法等等，这些计算机领域细小的枝叶的确引出了背后庞大而有趣的内涵。

这次作业从算法上讲并无难度，因为课上已经讲得很清楚了，而无出链和自环的情况也已经由平滑系数Alpha解决，剩下唯一有一定难度的问题就只有如何处理巨量的数据使其可以进行矩阵运算了。

单从所给的.docx文档入手，我发现所要求的迭代矩阵A满足

$$\mathbf{A} = \frac{1\text{-}\alpha}{\mathrm{N}}\boldsymbol{E} + \alpha\boldsymbol{S}$$

其中矩阵E是NxN的矩阵，这在所给数据集规模下是无法建立的。
但很快可以发现，S中绝大部分元素为0，即S是一个稀疏矩阵，既然只有百万数量级的有效信息，就一定有优化空间的方法。果然，在百度上搜索Python处理稀疏矩阵，马上就给出了一个不错的方案：使用scipy库和numpy库建立稀疏矩阵，满足所有矩阵运算的操作。接下来的任务就非常简单了，利用三元组建立coo_matrix，考虑到出链都是基于列向量转换成csc_matrix很容易就能利用本身自带的indptr列表确定各列的非零元数量，也就是对于网页Yi的出链总数ni，这样，只需要对邻接矩阵自身做列初等变换即可得到S矩阵。
可惜的是E矩阵仍然无法建立，它全是1，不是稀疏阵，因此不使用PRn+1 = A·PRn的做法，而退一步用PRn+1 = (1-alpha)/N·e+S·PR，迭代得到最终结果。这便是Python版本的实现方式。

然而事情并没有描述的这样一帆风顺，在最初我并没有找到教我用三元组建立稀疏矩阵的教程，因此采用的是C++中vector<>动态数组邻接表的做法，由于不使用线性代数的知识，只能回到最初的公式，对于每个Yi网页，都向它所指的X网页提供权值，最后给每个网页加上随机访问概率，实现一次迭代。得益于C++的高性能，算法实现比较顺利。

最终的结果是Python和C++的两个版本代码实现输出的PR值都能对应上，而小规模数据下，二者的输出结果与网上流传的代码一致，主观上来看，最终是成功完成了任务。

不得不说Python在各类库的加持下效率很高，非常强大。即得即用的库既能方便写代码的人调用，也能优化代码效率，库背后的算法思想也写得十分清晰，比起C++更为亲民。然而在使用过程中我也发现这种模式的问题，版本更迭导致代码失效是个很头痛的问题。诸如此类的问题在Python本身也有体现——Python 2.7到Python 3的版本跨越使得部分互联网上流传的老旧代码无法使用，而pygraph库的安装、使用也与版本息息相关，导致许多网上流传的代码无法运行。更有甚者，在Python3.9版本下正常运行的代码无法在Python3.10上运行，官方对于模板名称的朝令夕改使得部分代码的维护变得比较困难。从这个角度看，C/C++似乎在标准体系的加持下更可靠，但这种对老版本老标准的长期维护也形成了对新标准的打压——为了保证代码可维护性，绝大多数情况都是采用c97或c98等老标准编写代码，而c11/c17等新版本的特性既没有多少人去学习，也没有多少人去使用，以至于发出来的代码中新语法不被网友们接受，这也是一大问题。

说到最后有些偏题了，但最终我想表达的东西还是说，这堂课真的太有意义了。如果可以，我建议2022级珠峰班的同学也可以上这门课。我知道这门课对于同学们有一定的编程能力要求，这一部分缺憾可以通过把程序设计课移到大一上来补足，但我确信程序设计课能讲授的东西不足以完全搞定这门课，因此可以督促同学们去自学，去探索新知识，去了解、掌握更多课外的东西，可能这些“困难”和“没时间完成的任务”才是最能提高我们的东西。

——孙权恩202108101013