

Patrick Kelly

COMPSCI 446

10/1/2021

Page Rank Project

My code for the Page Rank algorithm is on lines 38 to 52 in pagerank.py. All that was used in this project in the standard Python library. The first thing the program does is loop through the "links.srt" file line by line and splits each line into two strings. The first string is the source and the second string is the target. In the same loop the program also initializes i and r which can be found in the book's procedure. Then the program gets the page rank algorithm which follows the book's procedure closely. Instead of using a nested loop my program uses an accumulator for the extra pages that have no outgoing edges. After the page rank algorithm the program writes the counts for pagerank and inlinks to two different text files. The program has larger space complexity in the pagerank algorithm because of the accumulator. The procedure from the text book is $O(n^2)$ but when you use the accumulator the run time is $O(n)$. The accumulator allows each Page Rank to be added to its respective page outside of the other loop. Pages with the most inlinks and the pages with the highest pagerank are relatively similar early in the list however as you go down the list more differences arise. These differences could arise from different quality of links on pages. If the PageRank scores were initialized to random values, this theoretically should not matter too much given a large enough data set, however with a small data set it could really throw off the PageRanks. If the 'random surfer' is removed, meaning the λ is zero it will only follow links on the page instead of randomly jumping to

new links. In my program this causes the pages that appear in the top 75 Page Ranks to be similar to the actual top 75 there are still many differences and Page Rank scores are usually higher.