**Steps Followed:**

I divided this assignment into 4 chunks

1. Solr setup:
   1. Downloaded Solr version 7.7.3
   2. Unpacked the package
   3. Started the solr server using bin/solr start
   4. Created a new core using command bin/solr create -c myexample
   5. Downloaded the LA Times data from Google drive
   6. Indexed the webpages using Solr by issuing the command: bin/post -c myexample -filetypes html /Users/esha/Documents/CSCI 572/hw4/latimes
   7. Open Solr UI at http://localhost:8983/solr/
2. Java Code for Creating Edge List
   1. Downloaded the JSoup jar file and added it as an external JAR file to my project.
   2. Used the URLtoHTML\_la\_times.csv and the crawled la times data created the edgelist.txt which is essentially a graph of all the in and out links of a page.
3. Python Code for Computing Page Rank
   1. Used the networkx package to compute the pagerank score of each page using the edge list file created in the last step.
   2. The parameters I used were: alpha=0.85, personalization=None, max\_iter=30, tol=1e-06, nstart=None, weight='weight',dangling=None
4. PHP code for creating the UI
   1. Created the front end in PHP
   2. It is responsible for taking user input in a search form. Along with this the user has an option (radio button) of using “Lucene” or “Page Rank” to sort the search results.
   3. Once the user selects the search option and submits the search query, it is sent to the solr server.
   4. Solr returns the results of the search query
   5. We display the top 10 results of the search query on our webpage.
   6. The details we display are title, id, description and url
   7. The url and title are clickable (enclosed in a href tags). If the user clicks on them they are taken to the clicked result’s webpage.

**Why some pages have higher PageRank values than others:**

Some pages have higher PageRank value than others because page rank is calculated using the number of in links to the page and the number of out links going out of the page. We can see that even though links like <https://www.latimes.com/california>, <https://www.latimes.com/opinion>, <https://www.latimes.com/politics> are not very relevant to our search queries since they serve as “home pages” they are displayed quite high in the search results. This is because they have a lot of external pages pointing to them as they are some of the main pages of the LA Times.

**Flow Diagram:**

Image 1:

Graphical user interface, text, application

Description automatically generated

Image 2:

Timeline

Description automatically generated

Image 3:Timeline

Description automatically generated

Image 4:

Clicking on the top link of Image 3 opens the web page

Graphical user interface, website

Description automatically generated

**Table Of Query Search Results:**

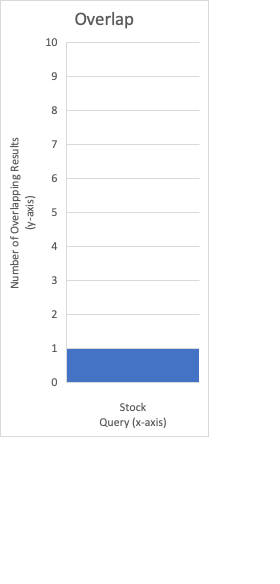
Table

Description automatically generatedTable

Description automatically generated with low confidenceA picture containing table

Description automatically generated

**Overlaps:**



The only Query to have an overlap was “Stock” which had one overlapping result between Lucene and the PageRank algorithm.

For all the rest of the queries we had 0 overlapping results.