COMP2521 (19T2)

Ass2 (part-1) : How to Implement?

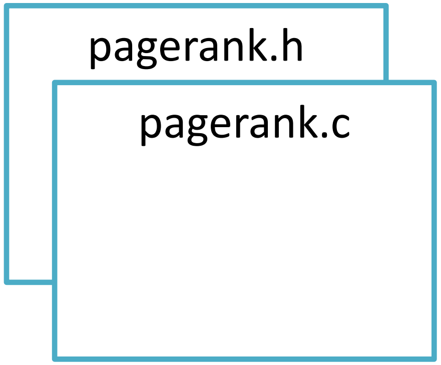
Notes:

- The document offers some **suggestions only**, with incomplete pseudo code
- The pseudo code is easy to read, but may not be efficient. You need to improve it!
- You can use code from labs/lecture material, however, must acknowledge it and provide a reference. For example you can use graph ADT implementation from one of the labs, and adapt it for this assignment.
- You can build a graph structure using Adjacency Matrix or List Representation.

readData.c

```
List_of_Urls ← GetCollection( )
    Create a set (list) of urls to process by reading data from file “collection.txt”

Graph g ← GetGraph(List_of_Urls )
    Create empty graph (use graph ADT in say graph.h and graph.c)
    For each url in the above list
        • read <url>.txt file, and update graph by adding a node and outgoing links
```



pagerank.h

pagerank.c

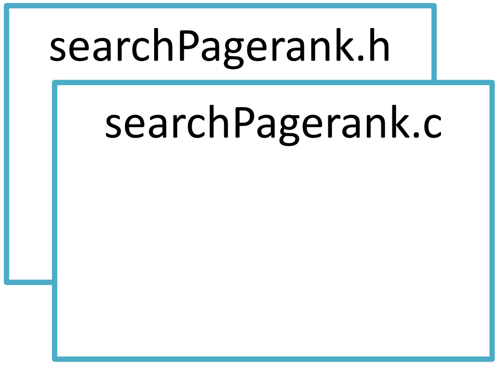
pagerank.c

```
Get args : d, diffPR, maxIterations

List_of_Urls ← GetCollection( )
Graph g ← GetGraph(List_of_Urls)

List_Urls_PageRanks = calculatePageRank(g, d, diffPR, maxIterations );
Ordered_List_Urls_PageRanks  = order (List_Urls_PageRanks )

Output Ordered_List_Urls_PageRanks to “pagerankList.txt”
```



searchPagerank.h

searchPagerank.c

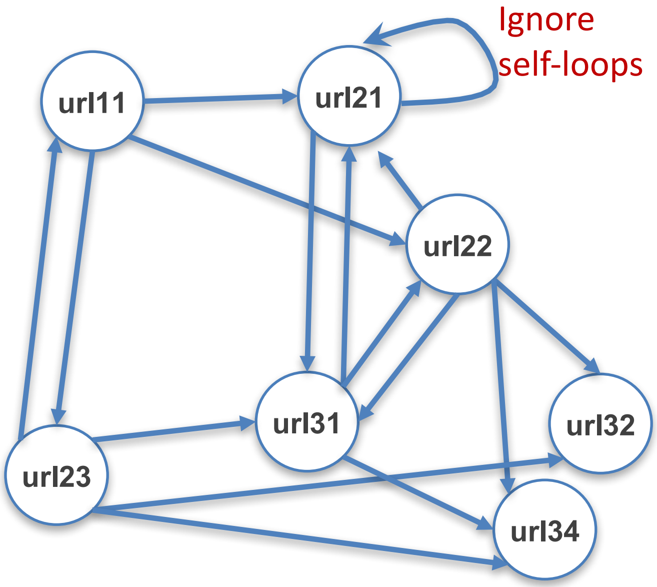
searchPagerank.c

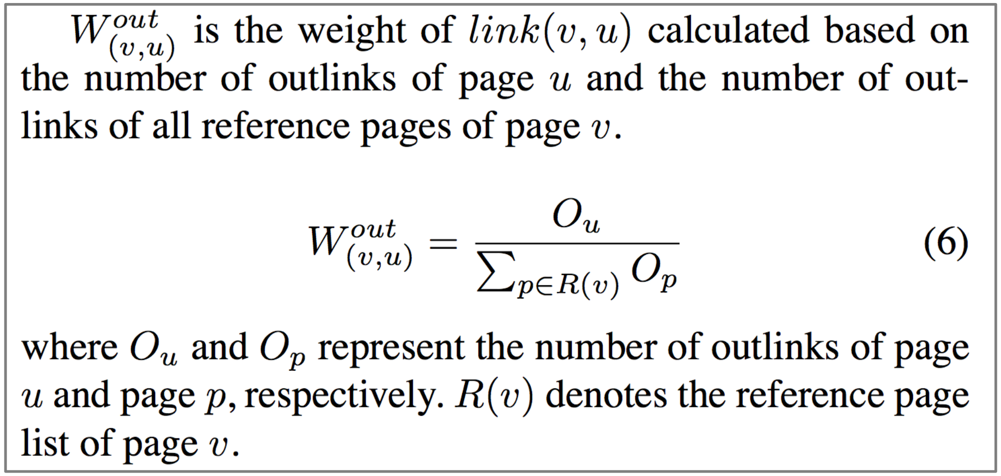
Get query words from arguments

```
matched_Url_list ← findMatchedUrls(“invertedIndex.txt”, queryWords)
matched_Urls_with_PR ← findPagerank(“pagerankList.txt”, matched_Url_list)
```

Output ordered urls on stdout

W^{out} : How to calculate?

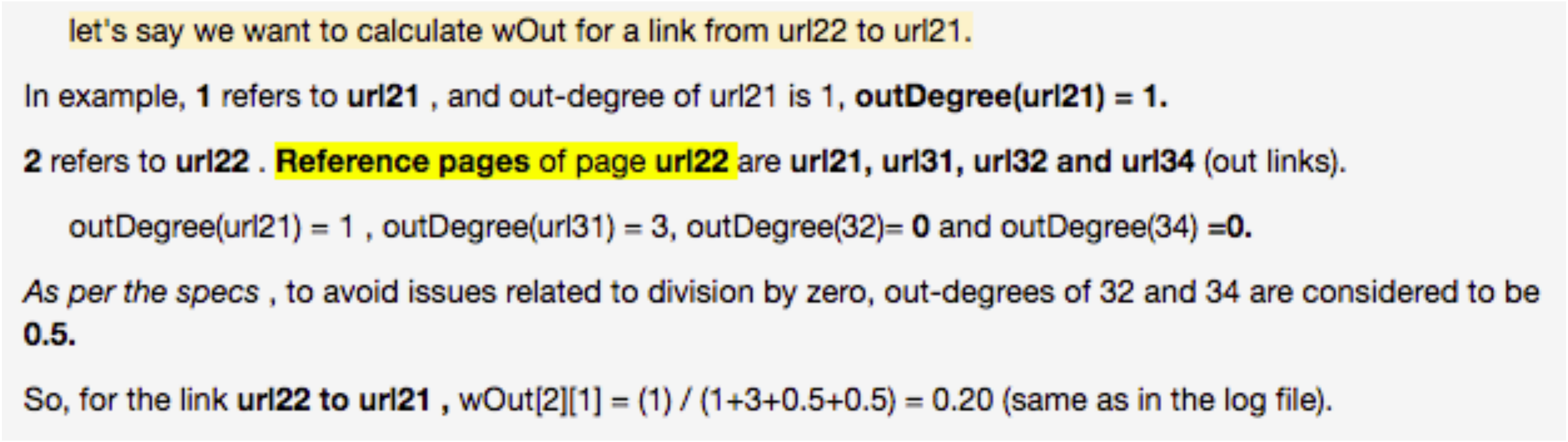



$W^{out}_{(v,u)}$ is the weight of $link(v,u)$ calculated based on the number of outlinks of page u and the number of outlinks of all reference pages of page v.

$$W^{out}_{(v,u)} = \frac{O_u}{\sum_{p \in R(v)} O_p} \qquad (6)$$

where O_u and O_p represent the number of outlinks of page u and page p, respectively. $R(v)$ denotes the reference page list of page v.

- **Reference pages** of url22 are url21, url31, url32 and url34
- **Ignore** self-loops (i.e. url21 to url21), and also parallel edges (if exist)


let's say we want to calculate wOut for a link from url22 to url21.

In example, **1** refers to **url21** , and out-degree of url21 is 1, **outDegree(url21) = 1.**

2 refers to **url22** . **Reference pages of page url22** are **url21, url31, url32 and url34** (out links).

outDegree(url21) = 1 , outDegree(url31) = 3, outDegree(32)= **0** and outDegree(34) =**0.**

As per the specs , to avoid issues related to division by zero, out-degrees of 32 and 34 are considered to be **0.5.**

So, for the link **url22 to url21 ,** wOut[2][1] = (1) / (1+3+0.5+0.5) = 0.20 (same as in the log file).