CS 6200 Project 1: PAGE RANK IMPLEMENTATION

Name: Ajay Pandit /MS in Computer Science Fall '12/ CCIS Username: ajay / Email: pandit.aj@husky.neu.edu

----

(Project: Part A) Implement the iterative PageRank algorithm as described above. Test your code on the six-node example using the input representation given above. Be sure that your code handles pages that have no in-links or out-links properly. (You may wish to test on a few such examples.) Your task will probably be easier if you don't require loading the entire link graph into memory.

To hand in: List the PageRank values you obtain for each of the six vertices after 1, 10, and 100 iterations of the PageRank algorithm.

(Ans) The Implementation for the given algorithm is given below:

```
############################### IMPLEMENTATION #####################################
# Importing the math library
import math

# taking the sample graph as input
input_file = "in-links-file.txt"

# defining dictionary M which is a combination of a page and set of pages that link to that
page
M = {}

# defining dictionart L which is a combination of a page and number of outlinks the page has
L = {}

# defining list P which is a list of all the nodes in the graph
P = []

# defining list S which is a list of sink nodes (nodes which 0 outlinks)
S = []

# defining list page_rank which stores the page rank of the pages
page_rank = {}

##################################################################################

#populating all the data structures with the required values which will be used for computing
the Page Rank
def initialize_lists_with_data (get_input_from_file, in_links_set, out_links_count, Pages,
sink_nodes):
    get_input_from_file = open(input_file, "r")
    for eachline in get_input_from_file:
        eachline = eachline.strip()
        nodes = eachline.split(" ")
        page = nodes[0]
        Pages.append(page)                              # initialized P with list of nodes
        in_links_set[page] = tuple(nodes[1:])           # initialized M with page and its
links

#    print "Page List = ", Pages
#    print "<Page: Links connected> = ", in_links_set

    for key,value in M.iteritems ():
        for node in value:
            temp = 1
            if node in out_links_count:
                temp = out_links_count[node]
                temp += 1
            out_links_count[node] = temp                # initialized L with page and no of 
outgoing links

#    print "<Page: OutLink Count> = ", out_links_count

    for value in Pages:
        if value not in out_links_count:                # checking whether the Value field of 
the dictionary is blank?
            sink_nodes.append(value)                    # if yes add to list of sink nodes

#    print "Sink Nodes = ", sink_nodes
    print "Initialization Completed Successfully"

##################################################################################

def get_page_rank (page_rank, P, L, S, M, d, N, no_of_iterations):

    k = 1/float(N)
    print "N = ", N
    for p in P:
        page_rank[p] = k                  # initial value
#    print "Initial Page Rank = ", page_rank

    new_page_rank = {}
    prev_perplexity = 0.0
    i = 0
    j = 0
    print "\nPrinting Perplexity Values till it converges:"
    while i < no_of_iterations:
        perplexity = get_perplexity(page_rank)
        if abs (int(perplexity) - int(prev_perplexity)) == 0:
            j += 1
        else:
            j = 1
        if j == 5:
            break
        prev_perplexity = perplexity
        print "Perplexity = ", prev_perplexity
        sinkPR = 0
        for s in S:
            sinkPR += page_rank[s]         # calculating total sink PR
        #print "sinkPR = ", sinkPR
        for p in P:
            new_page_rank[p] = (1-d)/N + d*sinkPR/N
            for q in M[p]:
                new_page_rank[p] += d*page_rank[q]/float(L[q])
        for p in P:
           page_rank[p] = new_page_rank[p]
           #print "page rank = " , page_rank[p]
        i += 1

    print "Page-Rank Calculation Completed Successfully"
    return page_rank

################################################################################
```

```
#calculating the perplexity value based on the given formula
def get_perplexity(page_rank):
    check_convergence = False

    perplexity_value = 0
    entropy_value = 0

    for rank in page_rank.values():
        if rank != 0:
            entropy_value += rank*math.log(1/float(rank),2)

    perplexity_value = pow(2, entropy_value)
    return perplexity_value

################################################################################

#printing the top 50 pageranks
def printing_top_pageranks(final_page_rank):
    # Sorting the list by the Page Rank
    print "\nPrinting top page ranks"
    i = 0
    for key, value in sorted(final_page_rank.iteritems(), key = lambda (k,v): (v,k), reverse = 
True):
        if i == 50:
            break
        else:
            print "%s: %f" % (key, value)
        i += 1

####################################################################################

#printing the top 50 pages with inlinks
def printing_top_inlinks(M):
    keys_list = []
    in_links_count = []
    res = []

    key_list = list (M.keys())
    in_links_count = list (M.values())

    for s in in_links_count:
        res.append (len(s))

    rank_inlinks = dict(zip(key_list,res))
    #print count
    print "\nPrinting top pages with inlinks"
    p = 0
    for key, value in sorted(rank_inlinks.iteritems(),key=lambda (k,v): (v,k), reverse=True):
        if p==50:
            break
        else:
            print "%s: %d" % (key,value)
        p += 1

####################################################################################

# Calling the initialize function for filling the data structure with the required values
initialize_lists_with_data (input_file, M,L,P,S)
```

```
# calculating the number of pages
N = len (P)

# d is the PageRank damping/teleportation factor; use d = 0.85 as is typical
d = 0.85

#setting up the number of iterations
no_of_iterations = 100

# calculating the final page rank after passing the no of iterations as last parameter
final_page_rank = get_page_rank (page_rank, P, L, S, M, d, N, no_of_iterations)

#printing the top 50 pages by rank
printing_top_pageranks(final_page_rank)

#printing the top 50 pages by number in-links
printing_top_inlinks(M)

####################################################################################
```

Running the Code instructions:

1. For initializing the data lists and dictionaries used within the program kindly use the function: ***initialize_lists_with_data*** and pass the required parameters

2. For the getting the Page Ranks we need to call the ***get_page_rank*** function and then pass the filled lists and dictionaries and certain constant values to obtain the required page rank

3. For getting the perplexity values you can again use the same function ***get_page_rank*** and this will keep on giving you the perplexity values unless they get converged (criteria: If the integer value of the perplexity repeats 4 times then we need to stop)

4. For printing the top 50 pages with highest page ranks use the function ***printing_top_pageranks***

5. For printing the top 50 pages with highest number of inlinks use the ***function printing_top_inlinks***

Please Note: Sometimes the entire process may take some amount of time depending upon the system configuration on which this is running

CS 6200 Project 1: PAGE RANK IMPLEMENTATION

Name: Ajay Pandit /MS in Computer Science Fall '12/ CCIS Username: ajay / Email: pandit.aj@husky.neu.edu

<u>Output after given number of iterations</u>:

1. After 1 Iteration

```
A: 0.249306
E: 0.213889
F: 0.143056
C: 0.143056
D: 0.131250
B: 0.119444
```

2. After 10 Iterations

```
A: 0.252036
E: 0.187107
F: 0.151294
C: 0.151294
B: 0.139307
D: 0.118963
```

3. After 100 Iterations

```
A: 0.255638
E: 0.186322
F: 0.150962
C: 0.150962
B: 0.138137
D: 0.117979
```

(Project: Part B) Download the in-links file for the WT2g collection, a 2GB crawl of a subset of the web. This in-links file is in the format described above, with the destination followed by a list of source documents.

Run your iterative version of PageRank algorithm until your PageRank values "converge". To test for convergence, calculate the perplexity of the PageRank distribution, where perplexity is simply 2 raised to the (Shannon) entropy of the PageRank distribution, i.e., 2H(PR). Perplexity is a measure of how "skewed" a distribution is --- the more "skewed" (i.e., less uniform) a distribution is, the lower its perplexity. Informally, you can think of perplexity as measuring the number of elements that have a "reasonably large" probability weight; technically, the perplexity of a distribution with entropy h is the number of elements n such that a uniform distribution over n elements would also have entropy h. (Hence, both distributions would be equally "unpredictable".)

Run your iterative PageRank algorithm, outputting the perplexity of your PageRank distribution until the perplexity value no longer changes in the units position for at least four iterations. (The units position is the position just to the left of the decimal point.)

For debugging purposes, here are the first five perplexity values that you should obtain (roughly):
183811, 79669.9, 86267.7, 72260.4, 75132.4
To hand in: List the perplexity values you obtain in each round until convergence as described above.

(Ans) Using the same program mentioned above we can calculate the perplexity values

Output:
The Perplexity values are shown as below until it converges:

```
Perplexity =  183810.999998
Perplexity =  79669.9231957
Perplexity =  86267.6741024
Perplexity =  72260.3536067
Perplexity =  75132.4076593
Perplexity =  68932.6029131
Perplexity =  71197.8334108
Perplexity =  67782.5377846
Perplexity =  69379.5774141
Perplexity =  67383.7075589
Perplexity =  68477.8018835
Perplexity =  67207.1847963
Perplexity =  68004.1538837
Perplexity =  67138.9553795
Perplexity =  67708.2593908
Perplexity =  67131.6639346
Perplexity =  67524.4769137
Perplexity =  67132.1110911
Perplexity =  67413.7101219
Perplexity =  67138.8498145
Perplexity =  67339.825439
Perplexity =  67149.7850062
Perplexity =  67290.830658
Perplexity =  67158.7620791
Perplexity =  67259.2257455
Perplexity =  67166.0293807
Perplexity =  67237.7802261
Perplexity =  67172.3205073
Perplexity =  67223.127439
Perplexity =  67177.144373
Perplexity =  67213.3194519
Perplexity =  67180.7511381
Perplexity =  67206.5975774
Perplexity =  67183.5491212
Perplexity =  67201.9331033
Perplexity =  67185.6323888
Perplexity =  67198.7420932
Perplexity =  67187.1593956
Perplexity =  67196.5257186
Perplexity =  67188.2983336
Perplexity =  67194.9793762
Perplexity =  67189.1321605
Perplexity =  67193.9036297
Perplexity =  67189.7403781
Perplexity =  67193.1507383
Perplexity =  67190.1847357
Perplexity =  67192.621374
Perplexity =  67190.5078833
Perplexity =  67192.250355
Perplexity =  67190.7423803
Perplexity =  67191.9887961
Perplexity =  67190.912902
Perplexity =  67191.804427
Perplexity =  67191.036087
Perplexity =  67191.6743361
Perplexity =  67191.1253299
```

The perplexity converges at 67191 as seen from the above list of values

CS 6200 Project 1: PAGE RANK IMPLEMENTATION

Name: Ajay Pandit /MS in Computer Science Fall '12/ CCIS Username: ajay / Email: pandit.aj@husky.neu.edu

----

(Project: Part C) Sort the collection of web pages by the PageRank values you obtain.

To hand in: List the document IDs of the top 50 pages as sorted by PageRank, together with their PageRank values. Also, list the document IDs of the top 50 pages by in-link count, together with their in-link counts.

(Ans) Please find below the top 50 documents based on PageRank and number of Inlinks. The output can obtained by running the program mentioned above

Printing top 50 pages			
Based on Page Rank values **(Document ID: Page Rank)**		**Based on number of in-links** **(Document ID: In-links Count)**	
Document ID (page)	PageRank	Document ID (page)	Inlinks Count
WT21-B37-76	0.002679	WT21-B37-76	2568
WT21-B37-75	0.001526	WT18-B29-37	2269
WT25-B39-116	0.00147	WT01-B18-225	2260
WT23-B21-53	0.001372	WT23-B27-29	1940
WT24-B40-171	0.001245	WT21-B37-75	1704
WT23-B39-340	0.00124	WT27-B34-57	1257
WT23-B37-134	0.001205	WT27-B32-30	1255
WT08-B18-400	0.001144	WT08-B19-222	1041
WT13-B06-284	0.001125	WT08-B18-400	1011
WT24-B26-46	0.001085	WT10-B36-88	946
WT13-B06-273	0.001045	WT10-B36-90	943
WT01-B18-225	0.000988	WT10-B36-103	939
WT04-B27-720	0.000936	WT10-B36-89	896
WT23-B19-156	0.000894	WT21-B40-447	779
WT04-B30-12	0.000816	WT18-B28-345	728
WT24-B26-10	0.000807	WT12-B40-248	686
WT25-B15-307	0.000804	WT24-B26-2	625
WT07-B18-256	0.000775	WT25-B15-307	614
WT24-B26-2	0.000771	WT27-B28-203	598
WT14-B03-220	0.000716	WT18-B40-82	576
WT24-B40-167	0.000707	WT21-B37-71	560
WT14-B03-227	0.000685	WT22-B38-403	544
WT18-B31-240	0.00066	WT08-B01-173	539
WT04-B40-202	0.000659	WT13-B15-160	484
WT08-B19-222	0.000643	WT23-B30-88	478
WT27-B28-203	0.000627	WT18-B29-36	477
WT13-B15-160	0.000621	WT27-B28-177	470
WT13-B39-295	0.000617	WT13-B06-284	454
WT12-B30-56	0.000602	WT13-B06-273	454

WT10-B02-288	0.000576	WT07-B02-55	449
WT22-B38-403	0.000575	WT13-B39-295	443
WT14-B36-337	0.000558	WT17-B34-499	442
WT27-B34-57	0.000555	WT17-B34-500	436
WT23-B20-363	0.000551	WT24-B04-192	430
WT23-B01-40	0.00055	WT14-B36-337	417
WT27-B32-30	0.00055	WT17-B34-505	410
WT21-B40-37	0.000548	WT10-B33-300	409
WT21-B35-155	0.00054	WT23-B19-156	406
WT08-B08-60	0.000536	WT23-B31-215	402
WT04-B22-268	0.000533	WT17-B34-503	402
WT14-B02-400	0.000533	WT23-B23-51	400
WT18-B14-66	0.000532	WT08-B11-28	396
WT23-B27-31	0.000526	WT23-B12-215	388
WT23-B38-120	0.000521	WT23-B01-107	384
WT06-B35-151	0.00052	WT23-B30-105	380
WT06-B14-69	0.000519	WT17-B34-506	376
WT06-B35-161	0.000518	WT17-B34-504	374
WT10-B33-300	0.000517	WT17-B34-498	374
WT14-B36-336	0.000515	WT14-B36-323	373
WT14-B36-335	0.000515	WT07-B23-234	371

CS 6200 Project 1: PAGE RANK IMPLEMENTATION

Name: Ajay Pandit /MS in Computer Science Fall '12/ CCIS Username: ajay / Email: pandit.aj@husky.neu.edu

----

(Project: Part D)Examine the top 10 pages by PageRank and in-link count in the Lemur web interface to the collection by using the "e=docID" option with database "d=0", which is the index of the WT2g collection. For example, the link

http://fiji4.ccs.neu.edu/~zerg/lemurcgi_IRclass/lemur.cgi?d=0&e=WT04-B22-268

will bring up document WT04-B22-268, which is an article on the Comprehensive Test Ban Treaty.

To hand in: Explain why these documents have high PageRank values, i.e., why is it that these particular pages are linked to by (possibly) many other pages with (possibly) high PageRank values. Are all of these documents ones that users would likely want to see in response to an appropriate query? Which one are and which ones are not? For those that are not "interesting" documents, why do they have high PageRank values? How do the pages with high PageRank compare to the pages with many in-links? In short, give an analysis of the PageRank results you obtain.

(Ans)

Examining documents based on the highest number of inlinks:

Please find below the top 10 pages with highest number of inlinks

Document ID	**Number of Inlinks**	**Page Contains**
WT21-B37-76	2568	The Economist Homepage
WT18-B29-37	2269	Environmental News State
WT01-B18-225	2260	Online Library of Drug Policy
WT23-B27-29	1940	Welcome to the SportsGate
WT21-B37-75	1704	The Economist : Copyright Notice
WT27-B34-57	1257	Skeptics Society Message Board
WT27-B32-30	1255	WWWBoard Frequently Asked Questions and Answers
WT08-B19-222	1041	Agreement and Policy Usage
WT08-B18-400	1011	General Disclaimer
WT10-B36-88	946	A site for the gay, lesbian, bisexual and transgender community

Based on the above table we are able to conclude the following points:

- The first page *The Economist Homepage (WT21-B37-76)* is having the highest number of inlinks because if we visit any other page we always need to return to the homepage because that is the page which connects us to the different sections of the website. Every other page within a website has an outlink which directs us to the homepage else we would never be able to return to the home page. Also during search results we may directly jump to different sections but we shall always know which is the website which is providing us the information. Also during every search it is the homepage that always comes up which leads to more people visiting them and since homepage has links to other sections it has a very high page rank
- The second page *WT18-B29-37* page provides information regarding the Environmental news which is also linked to the news page and since news page has a high rank it increases the chances of this page getting visited and hence the page rank for this page is also high. This page provides regional information

and all the top stories are listed first which makes it more interesting for the people who are mainly interested in news that are more popular. Also it is usual that such news pages always has a high page rank

- The third page *WT01-B18-225* is the Online Library for Drug policy is a search page where people visit and search the drug for which they are looking for policies to use. Such website are mostly linked to the health department and are visited more often to provide the actual policies for usage of the different drugs, hence such pages are visited more often by the people which has resulted in getting a high page rank for this page. Also since this provides the policy other online library for drug pages are linked to this page which increases the inlinks count
- The fourth page *WT23-B27-29* is the Welcome to SportsGate which is a kind of page where people can link their web pages to it and get themselves visible to pages which has a high page rank because if we need to visit pages that are linked within the SportsGate then we need to go through this page and this increases the probability of people visiting them more. This page provides other web pages a chance to have a higher visit rate and for which they charge a premium amount which is also mentioned on the page
- The fifth page *The Economist: Copyright Notice (WT21-B37-75)* is having a significant number of inlinks. This page mainly consists of the copyright information which mainly prevents people from copying any kind of material from the website and posting them for personal or commercial use. This page also has a high page rank and the number of inlinks to remind the users of the website that they can only read or view the contents and not copy anything without prior permission and incase they need to publish any kind of information they need to agree to the terms and conditions
- The sixth page *WT27-B34-57* is a Skeptics Society Message Board which contains a forum where people ask their questions and get feedback over them. This page is also important since people from different departments come together to answer the questions that are raised by the people from the same or different department. The replies can be viewed by other people so that the same kind of questions are not repeated. Such pages usually have a high number of inlinks are Q&A are very common for the websites and as it is linked to most of the pages which already have a high page rank
- The seventh page *WT27-B32-30* contains the WWW Board frequently asked questions which is usually visited by lot of people to get all kinds of information. This page has inlinks from most of the other pages which makes this as a very popular link and people tend to click and visit this page. Also which visiting other pages they may get some questions in mind which makes them visit this page more often
- The eight page *WT08-B19-222* contains the agreement policy and usage policy of the website. Since these days all the websites provide such kind of agreement policies at the time of visiting them. It contains a high page rank since it has lots of inlinks from other pages which have to make people accept agreements. Youtube always shows the agreement of 18+ yrs for videos it feels are adults
- The ninth page *General Disclaimer (WT08-B18-400)* also has a high number of inlinks as it says to the user of the website whether there are any kind of protection being offered by the website to the users of it. Such disclaimers tell the users that the websites are not responsible for any kinds of loss of sensitive information from their personal computers while they are using this website. Hence the user needs to be made aware of all the precautions that needs to be taken before visiting hence they are usually included in most of the pages so that they cannot be held responsible for any kind of loss of confidential information
- The tenth page *WT10-B36-88* is just a website specific for the community of gay and lesbians. This page is not important but still it contains inlinks for other pages which already have a high page rank as a result this page also gets visited often and hence has attained a higher page rank

CS 6200 Project 1: PAGE RANK IMPLEMENTATION

Name: Ajay Pandit /MS in Computer Science Fall '12/ CCIS Username: ajay / Email: pandit.aj@husky.neu.edu

Examining documents based on the page rank

Document ID	Page Rank	Page Content
WT21-B37-76	0.002679	The Economist Homepage
WT21-B37-75	0.001526	The Economist : Copyright Notice
WT25-B39-116	0.00147	Security Assurance Requirements Page
WT23-B21-53	0.001372	Web Development Team
WT24-B40-171	0.001245	The Evening news On-Line archive
WT23-B39-340	0.00124	Financial Reports
WT23-B37-134	0.001205	Important Information
WT08-B18-400	0.001144	General Disclaimer
WT13-B06-284	0.001125	Website Credits Information
WT24-B26-46	0.001085	Milton's homepage

Based on the above table we are able to conclude the following points:

- The first page *The Economist Homepage (WT21-B37-76)* is having the highest number of inlinks because if we visit any other page we always need to return to the homepage because that is the page which connects us to the different sections of the website. Every other page within a website has an outlink which directs us to the homepage else we would never be able to return to the home page. Also during search results we may directly jump to different sections but we shall always know which is the website which is providing us the information
- The second page *The Economist: Copyright Notice (WT21-B37-75)* is having a significant number of inlinks. This page mainly consists of the copyright which mainly prevents people from copying any kind of material from the website and posting them somewhere else. This page also has a high page rank and the number of inlinks to remind the users of the website that they can only read or view the contents and not copy anything without prior permission
- The third page *Security Assurance Requirement (WT25-B39-116)* does not contain any information still it contains a high page rank. The main reason this document is having such a high page rank it because it has a link directly from the main page which increases the probability of people visiting this page more often to check its contents
- The fourth page *Web Development Team (WT23-B21-53)* provides information about the people responsible for the website and ways in which they can be contacted in case they need to report any kind of failure information. Also it lists hobbies and general interests of the individuals. Although it has a high page rank we don't think that it will be useful for many people. It may be possible that all people must be provided due credit for the work and this information is mentioned under the copyright information as a result this page has been linked which has given it a high rank
- The fifth page *The Evening news Online Archive (WT24-B40-171)* contains the evening news information. It contains all the news information which makes it contain lots of links. Since this page is kind of central location for getting the news it connects to lots of other news information links. Also searching news is a common activity across the internet this page has been given a higher page rank
- The sixth page *Financial Reports (WT23-B39-340)* contains the links for the financial reports for all the companies in alphabetical order. This page has a higher rank because it is accessed by most of the people even inside the company and acts as a reference. Also any kind of reports are considered important from search perspective hence it may have been given a higher page rank
- The seventh page *Important information (WT23-B37-134)* contains a important disclaimer for the materials provided by health department. Maybe the pages with WT23 have links to such pages and mostly are a part of the health department. These days it is more often seen that articles carry a disclaimer and hence carries a high page rank

- The eight page *General Disclaimer (WT08-B18-400)* is just an information which may not be that important yet this has a higher page rank and inlinks. This node is a sink node. I feel that it can be a general practice to include such documents which may have made it to carry a high page rank. Almost all the websites today carry such general disclaimer
- The ninth page *Website Credits Information (WT13-B06-284)* gives information about different designation that people are holding as a part of the Web Development team. May be this information is linked with the Web Development Team page which also contains similar kind of information along with a high page rank. This information may be very useful within the entire team to report any kind of issues or problems that the website is facing which can then be fixed as soon as possible
- The tenth page *Milton's homepage (WT24-B26-46)* is regarding Dr. Milton who is a lecturer in the department of psychiatry at University of Michigan. It may have been possible that the sequence of documents WT24 contains pages related to psychiatry which may have resulted in this page having high page rank

CS 6200 Project 1: PAGE RANK IMPLEMENTATION

Name: Ajay Pandit /MS in Computer Science Fall '12/ CCIS Username: ajay / Email: pandit.aj@husky.neu.edu

References:

Learning Python
http://www.daniweb.com/software-development/python/114
http://www.sthurlow.com/python/
http://www.stackoverflow.com/
http://www.greenteapress.com/thinkpython/html/book011.html

PageRank information
http://en.wikipedia.org/wiki/PageRank
http://pr.efactory.de/e-pagerank-algorithm.shtml
http://www.math.umass.edu/~law/Research/PageRank/Google.pdf
http://www.webworkshop.net/pagerank.html
http://www.markhorrell.com/seo/pagerank.html
http://ilpubs.stanford.edu:8090/750/1/2003-29.pdf

Perplexity
http://www.itl.nist.gov/iad/mig/publications/proceedings/darpa98/pdf/lm30.pdf
http://ilpubs.stanford.edu:8090/750/1/2003-29.pdf