

Use the python client that was give to you to generate a directory of 20K files with a max link count of 375. Copy the created files into a directory under a storage bucket on cloud google storage. Write a program in Python that opens the bucket, reads the list of files, reads each file and computes the following properties:

- Average, Median, Max, Min and Quintiles of incoming and outgoing links across all the files.
- Construct a graph of the pages, and compute their pagerank (PR) and output the top 5 pages by their pagerank score.
- Code the original iterative pagerank algorithm where $PR(A) = 0.15/n + 0.85(PR(T_1)/C(T_1) + \dots + PR(T_n)/C(T_n))$ where $PR(X)$ is the pagerank of a page X, $T_1..T_n$ are all the pages pointing to page X, and $C(X)$ is the number of outgoing links that page X has. Iterate until the sum of pageranks across all pages does not change by more than .5% across iterations.
- Write a test that verifies the correctness of your code and is independent of the randomly generated graph of 20K files.

Make sure your storage bucket is in us-central and you can run your program either on your laptop directly or on your google cloud shell instance. Make sure your bucket is world readable (give the right ACL permissions) so your program can be tested against your bucket by the TFs. In your report describe everything you had to do to set up the bucket and the permissions. Also include the output and timing of your program when running on your local computer and on your cloudbshell. Also include your total spend on cloud resources for the development and execution of this program with a screenshot of the relevant billing console view.

What to turn in: A pdf file containing the report outlined above, and including all relevant instructions on how to run and verify your code (i.e. your project name, your bucket name, the way to run your program, the meaning of any parameters it accepts etc). The program itself should be provided as a link to github containing your code. Make sure your github repository is accessible by the TAs and instructor and the link you provide works with git clone. If you use AI, state clearly how you used it in your writeup and explain your understanding of the information you received from it. If your code is complicated, we reserve the right to ask you to explain it and you should be able to do so without the assistance of an AI agent.

Good luck!