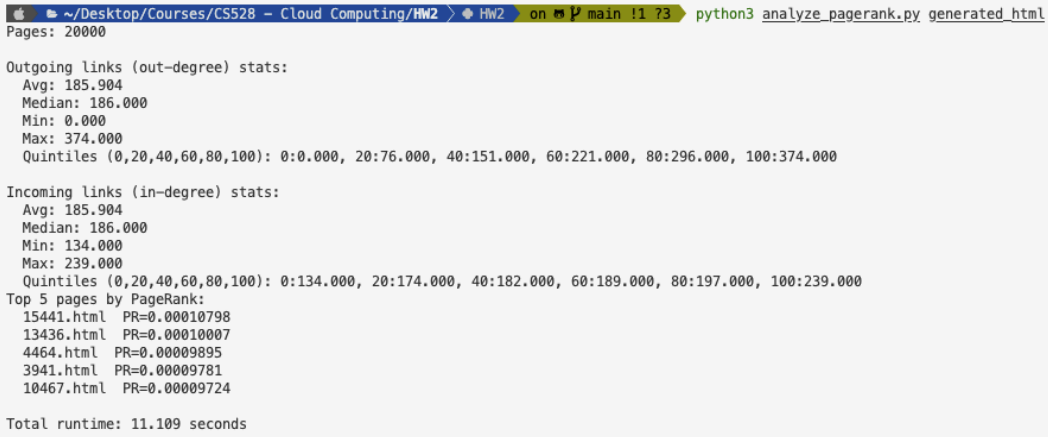
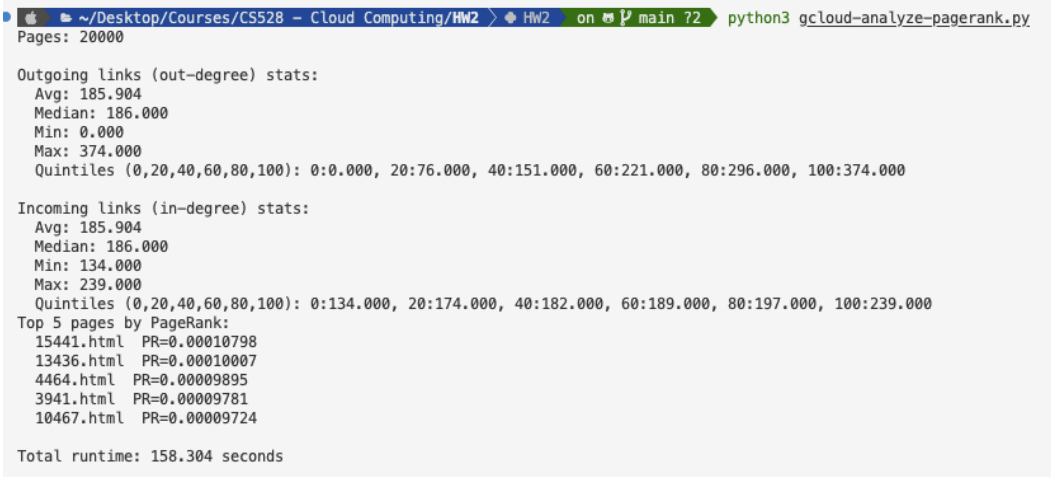
- Github link: https://github.com/IanTsai1/CS528---HW2
- Locally (**analyze_pagerank.py)**:
  - Steps:
    - “python3 generate-content.py -n=20000 -m=375”
    - “python3 **analyze_pagerank.py generated_html**”
      - The parameter references where the folder of the 20000 html is downloaded
  - Note:
    - In my code, I used ChatGPT which I have commented on and added my own explanation in the code file. I used AI to write up the code to find the quintiles, parse the graph, and write up the page rank algorithm. In those code, I also added my own explanation to ensure I fully understand the code.
  - 

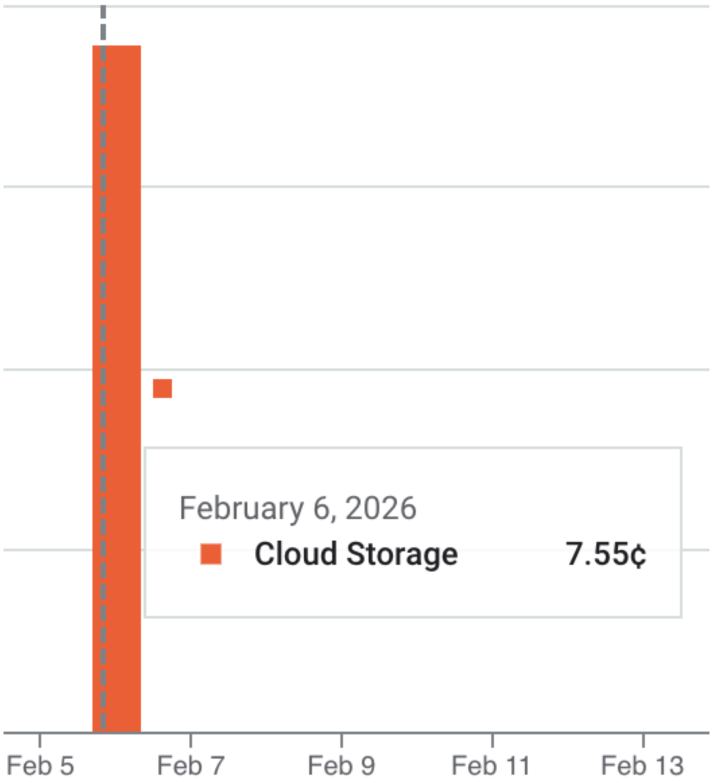
```
~/Desktop/Courses/CS528 - Cloud Computing/HW2 > HW2 on main !1 ?3 > python3 analyze_pagerank.py generated_html
Pages: 20000

Outgoing links (out-degree) stats:
  Avg: 185.904
  Median: 186.000
  Min: 0.000
  Max: 374.000
  Quintiles (0,20,40,60,80,100): 0:0.000, 20:76.000, 40:151.000, 60:221.000, 80:296.000, 100:374.000

Incoming links (in-degree) stats:
  Avg: 185.904
  Median: 186.000
  Min: 134.000
  Max: 239.000
  Quintiles (0,20,40,60,80,100): 0:134.000, 20:174.000, 40:182.000, 60:189.000, 80:197.000, 100:239.000
Top 5 pages by PageRank:
  15441.html  PR=0.00010798
  13436.html  PR=0.00010007
  4464.html  PR=0.00009895
  3941.html  PR=0.00009781
  10467.html  PR=0.00009724

Total runtime: 11.109 seconds
```

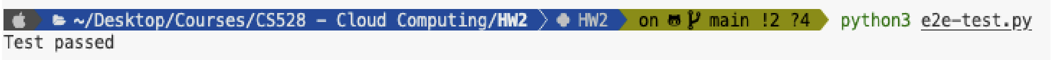
- Google cloud (**gcloud-analyze-pagerank.py)**:
  - How I ran the code:
    - “gcloud storage buckets create gs://iantsai-hw2 --location=us-central1”
    - “gcloud storage buckets add-iam-policy-binding gs://iantsai-hw2 --member=allUsers --role=roles/storage.objectViewer”
    - ‘gcloud storage cp -r generated_html gs://iantsai-hw2’
      - ‘Generated_html’ folder is where I stored all html files
    - ‘python3 **gcloud-analyze-pagerank.py**’
      - Ensure ‘pip install google-cloud-storage’ is installed
  - Project name = ‘cs528-485121’

- Bucket name = “iantsai-hw2”
- Note:
  - In my code, I used ChatGPT which I have commented on and added my own explanation in the code file. I mainly reused the code from ‘analyze-pagerank.py’ but I made modifications to ‘parse_graph_gcs’ to enable multithreading and to get files from gcloud bucket instead of local storage.
  - I used multithreading since accessing files from buckets are I/O bound and it takes a lot of time. By using multithreading, we can access data from other files during I/O bound.
- 

```
~/Desktop/Courses/CS528 - Cloud Computing/HW2 > HW2 on main ?2 > python3 gcloud-analyze-pagerank.py
Pages: 20000

Outgoing links (out-degree) stats:
  Avg: 185.904
  Median: 186.000
  Min: 0.000
  Max: 374.000
  Quintiles (0,20,40,60,80,100): 0:0.000, 20:76.000, 40:151.000, 60:221.000, 80:296.000, 100:374.000

Incoming links (in-degree) stats:
  Avg: 185.904
  Median: 186.000
  Min: 134.000
  Max: 239.000
  Quintiles (0,20,40,60,80,100): 0:134.000, 20:174.000, 40:182.000, 60:189.000, 80:197.000, 100:239.000
Top 5 pages by PageRank:
  15441.html  PR=0.00010798
  13436.html  PR=0.00010007
  4464.html  PR=0.00009895
  3941.html  PR=0.00009781
  10467.html  PR=0.00009724

Total runtime: 158.304 seconds
```

- Total spend:



- Verify code:
  - ‘python3 e2e-test.py’
  - To test if my code works, I create temporary html files that I know the page rank ranking. I use my pagerank and parse graph code from 'analyze_pagerank.py' and validate if the output from the file returns the expected output. Given that the temp html files are hard coded, I can find out the pagerank itself and then verify it. This is a e2e test on parse graph of html files and pagerank
  - This also works for ‘gcloud-analyze-pagerank.py’ as the only difference between ‘gcloud-analyze-pagerank.py’ and ‘analyze-pagerank.py’ is how the parse graph and file access works.
  - 

```
~/Desktop/Courses/CS528 - Cloud Computing/HW2 > HW2 > on main !2 ?4 > python3 e2e-test.py
Test passed
```