Padigala Lakshman Sai
2021201069

Directory Structure:

—2021201069
- |--indexer.py
- |--search.py
- |--stat.txt
- |--Report.pdf
- |--output (folder)

Output folder contains the following output after Indexing.

|--output (folder)

|--indexId.txt

This file is created for every 5000 files parsed and stored. These files are deleted after merging the index files.

|--finalIndexId.txt

This file is created after parsing all the docs completed and k way merge done.

|--titles.txt

This file contains all the titles and assigned doc_id's in each line.

|--title_offset.txt

This contains file pointers for the titles in titles.txt

|--secondary_index.txt

This contains the secondary index, meaning finalIndex doc_id and start word and last word of every finalIndex.txt

|--stat.txt

This contains the stats of index created, which will be used by search.py in the searching phase.

Optimization used:

Created secondary index file for making the search faster based on the index files created after k way merge.

Created the offset file for titles to extract the title of the article fastly after calculating tf-idf and ranking.

To avoid extra space usage, I have not considered zeros to avoid the sparse nature of the index files. Added only the fields which have counts greater than zero in the index.

Index Creation Time:

47000 seconds(~13 Hours) roughly on 91GB dump.

489.4 seconds on a smaller dump provided which is around 1.5GB.

Index Size:

24.64 GB on 91GB dump

388.2 MB for the smaller dump which is around 1.5GB

Format Of Final Index Created:

Eg: school:5426t2b3i1c4l3r1;4578t3b2l4r2; and so on…

Here school is the word indexed and it has been occurred in two docs with id's 5426 and 4578 with there count in doc 5426 is 14(2 in title, 3 in body, 1 in infobox, 4 in category, 3 in links and 1 in reference) and similarly in the next doc also.

indexer.py and search.py are the only code files

To run the code:

1.create a folder named 'output' in the folder 2021201069.

2. command for Indexing creation and merging:

>>> ***python "./2021201069/indexer.py" "data.xml"***

3.command for Searching from a text file with queries:

>>> **python "./2021201069/search.py" "queries.txt"**

queries_output.txt file is created in the output folder only in specified format which includes search time.