Data Structures PA1 Report

B07202020
Hao-Chien Wang

December 3, 2020

1 Usage

Use `./bin/pagerank <DIFF> <d>` to run the program. The input files are placed under `./input/` and the output files are generated under the main folder (`./`). The scripts are placed under `./src/` and can be compiled using `make`. You can also run the script `./runall.sh` to run for all input combinations. Use `make clean` to clean output files, executables and `*.o` files.

2 Data Structures and Algorithms

I wrote my program in c++. I defined 2 classes: `Page` and `FindPageRank`. The foremer stores the index of the page (for example, the index of `page499` is 499, saved as `int`), the number of outbranching links (`int`), pages it links to (`vector<int>`, storing indices of the pages) and its page rank (`double`). The latter reads the input, compute page ranks and output the requested files. It stores the following information:

1. Both input values `DIFF` and `d` (as `double`)
2. A wordlist (`map<string, set<int>*`) with the string being all the words mentioned in the pages and the set (that the pointer points to) saving the indices of the pages that mentioned the word.
3. An array of pages, saving all the `Pages` objects.

All the above are defined in `src/pagerank.h` and `src/pagerank.cpp`. The output is generated in the following steps:

1. Read input, create `Page` objects, generate wordlist and `reverseindex.txt` (done by the constructor of `FindPageRank`).
2. Iterate until the terminating conditions are met (done by `FindPageRank.iterate()`).
3. Output `pr_xx_yyy.txt` (done by `FindPageRank.printPR()`).
4. Read `list.txt`, search in wordlist according to it and output `result_xx_yyy.txt` (done by `FindPageRank.search()`).