CSC301 HW6

Alex Zhang

Feb 2023

Question 1

For the algorithm, readFile method enables transforming text file into matrix represeations. The following code blocks is creating the number of two-paths in adjacency list represeation. In the last part of the main function, the BFS is used to find recommander and writing the file out. I also have some comments on codes.

Question 2

For the complexity analysis, my code complexity contains different parts. The first part is reading the matrix I think the time complexity will be $O(m)$. The second part is recording the number of two-paths in adjacency list representation which the time complexity will be $O(nd^2 + nd)$. The third part is fidning the recommandation integer which the time complexity will be $O(nd \log d + n^2 + nm + 2nd)$. Overall the time complexity for this algorithm will be

$$O(n^2 + nd^2 + nd \log d + 3nd + nm + m)$$

Which I think in this case the upper bound should be $O(n^2)$ or $O(nd^2)$ depending on the number of d.

Question 3

(a)

When doing the A16.txt file, both adjacency list and adjacency matrix method have roughly same time. However, when doing A1024.txt file, for adjacency list method, the time doing is about 231 millisecond. The adjacency matrix method costs 94165 millisecond. This is a big differece. The adjacency list is faster. I think the reason is he adjacency matrix method includes a matrix multiplication method when recommending, which the time complexity is $O(n^{2.81})$. However, in adjacency list method based on my complexity analysis in question 2, it only costs $O(n^2)$ or $O(nd^2)$ which is way cheaper then matrix multiplication.

(b)

Based on rec1024.txt file, I find one recommendation which the first word "existence" is recommended to the fourth word "unsubstantiality". The intermediate word is the second word which is "inexistence". I think it is meaningful because we cannot say something exists if there is lack of specific evidence. Or it is not substantial which we can hardly say existence.