
CE2001/CZ2001 ALGORITHMS

Project 2: Graph Algorithms

1. Problem

You are given an undirected unweighted graph G , which represents a city's road network with n nodes being intersections/endpoints and m edges being roads. Among the n nodes, h of them are hospitals and you are interested in finding, for each node, the distance (i.e., the number of edges in the shortest path) from each node to the nearest hospital. h could take any value from 1 to n .

- (a) Design an algorithm for computing the distance from each node in G to its nearest hospital. Output the distance and the shortest path for each node to a file.
- (b) Design an algorithm to complete the task (a) but its time complexity should not depend on the total number of hospitals h . You could skip (b) if your algorithm in (a) already satisfies this complexity constraint.
- (c) In some circumstances (e.g., big disasters, fires, etc.), having the distance from each node to the nearest hospital is not good enough. Instead, we are interested in finding the distances to top-2 nearest hospitals from each node. Revise your algorithm in (b) to accommodate this new requirement. If revision is not possible, you are free to design a new algorithm.
- (d) Propose an algorithm that works generally for computing the distances from each node to top- k nearest hospitals for an input value of k .

2. Data Source

As a quick start and for demonstration purpose, you could use a small input graph obtained from a random graph generator (<http://graphstream-project.org/doc/Generators/Random-graph-generator/>).

Your designed algorithms are required to be tested on at least one of the large-scale real road networks downloaded from <https://snap.stanford.edu/data/index.html#road>.

For both random graphs and real road networks, you should generate the number and the locations of hospitals to be used as input.

3. Implementation

You are required to implement all algorithms designed in this project.

[Input] The input for tasks (a)-(c) is a graph with hospitals marked, and that for task (d) has an additional input of k . Your program should take two input files for the graph: *file1* stores the graph structure in the same format as the real road network file downloaded; and *file2* stores the hospital information. The format of *file2* is as follows: the first line is “# *num*”, giving the total number of hospitals; then each hospital is placed in a single line by showing its node ID only. A sample is given below:

4

3

10

19

34

[Output] Outputs of the program should be fed to a file, unless for demonstration purpose.

[Experiments] Your program should be tested on at least one random graph and one real road network. Note that the real road network is of large-scale (over 1 million nodes/edges) and your program should be able to handle this input scale (memory/space issues may arise). You are required to conduct empirical study to see the effects of h and k on the performance of various algorithms and conclude if the empirical findings are consistent with your theoretic analysis.

[Others] The interface should be user-friendly so that lab supervisor can test on arbitrary input graphs. The work has to be well documented and submitted to their lab supervisors for verification.

4. Design and Analysis of Algorithm

You are required to describe your algorithms clearly, prove their correctness, and analyze their asymptotic time complexity. The lab supervisor will evaluate your work based on the correctness and completeness of analysis of algorithm and originality of the proposed algorithms. All reference materials have to be stated clearly in the report. Students are reminded that plagiarism is a very serious academic offence. If any student may not knowingly intend to plagiarise, but that should not be used as an excuse for plagiarism. Students should seek clarification from lecturers or lab supervisors if they are unsure whether or not they are plagiarising the work of another person.