



THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

COMPSCI.220.S1.T - Algorithms and Data Structures

Assignment 2 – Graph Algorithms

Due: Tuesday, 2008-05-06, 20:30.

Available from: <http://www.cs.auckland.ac.nz/compsci220s1t/assignments/>

This assignment is worth **100 marks** representing $1/12$ of your total course grade.

Requirements

- (16 marks) Contribute at least 1 good quality question on graph algorithms, and answer and comment on at least 4 questions, on the PeerWise system at <http://peerwise.cs.auckland.ac.nz/CS220/>.
- (42 marks) You are given a file containing projects and workers who worked on that project. Your job is to find, given two workers, the minimum length of a chain of collaboration joining them. (Two famous examples are when projects are movies and workers are actors (“Kevin Bacon game”), and when projects are mathematical papers and workers are (co)authors (“Erdős numbers”).

Your program will take as input the file of projects and workers, and a given pair s and t of worker names (all given as command line arguments), and will output the minimum length of a path of collaborators joining s and t .

- (42 marks) This questions asks you to devise and implement a method to find a treasure in a maze and explain how to get to it.

We represent the maze by a graph in the usual way (edges are corridors, vertices are intersections, use adjacency lists), and are also given a start vertex s and target vertex t as part of the input. The program should output a path from s to t or report that no such path exists. The input file will contain several maze problems.

Notes:

- You should first make sure that your algorithm and implementation are correct! It is up to you to come up with test cases and the corresponding correct output. Usually, if a program contains an error, this can be detected on a small test case. You are allowed to share test cases with the class via the class forum.
- Next you should consider efficiency. You may want to use a hybrid of some algorithms in the textbook. You are welcome to use the code available from the resources page of the course website in any way you choose.
- Please see below for information on input and output formats.

Questions involving programming

- Java should be used for all programming questions unless special permission is obtained from the lecturer. You may freely use the classes available from the resources page of the course website. Template code may also be provided. You need not use it, nor stick to the template format, but it will probably save you time to do so.
- Your answer to each question should be a single .java file (containing all nonstandard classes you use). You may assume that the markers have access to all standard libraries and also the classes on the resource page, with the same directory structure as there.
- A sample input and output file for each question will be available. The markers may check the output with a text comparison program, so it must be in EXACTLY the right format.

Further details and hints

For Q2, the following format will be used. The `String.split()` method may be useful for processing input. The first line of the input file contains a single character, the separator. Each other line contains a project title, followed by the workers on the project. The strings are separated by the separator character, which does not appear in any of the strings.

One way to model this using a graph is to let each project be a vertex, and each worker be a vertex. To build the graph, it may be useful to use some implementation of the Set interface, such as `java.util.TreeSet` or `java.util.HashSet`.

For Q3, the input file will consist of the input for several problems, one after the other. The input for each problem is the adjacency lists representation of the graph, followed by a line containing the vertices s and t . You do not need to check that the adjacency lists representation really represents a graph. The file terminates with a single line containing "0", as usual.

Marking

Q1 will be marked by the lecturer and markers, using a simple grading scheme: 8 marks for the contributed question (all or nothing), and 2 for each answered and commented on.

For Q2 and Q3, half of the marks will be awarded for correctness, and half for speed.

The pass mark for the correctness test will be at least 90%. If you fail this test, you automatically score 0 for the speed test. The speed test marks are allocated according to the running time of the student answer compared to that given by the lecturer. Correctness is more important than speed, but speed (or at least avoiding slowness) is also important if you want to get a high grade.

Submission

The due date is Tuesday, 6 May 2008, 8:30 p.m. (ADB time) [then penalty linearly grows in time from 0% to 50% of marks until 2 days afterward; no submission allowed after that time]. Bonus also grows linearly in time from 0% to 10% of marks as time goes backwards by 5 days.

Submit your assignment to the Assignment Dropbox (ADB) system. You have to electronically hand in

- a single Java program `Distance.java` for Q2;
- a single Java program `FindPath.java` for Q3.