Theory of Computing

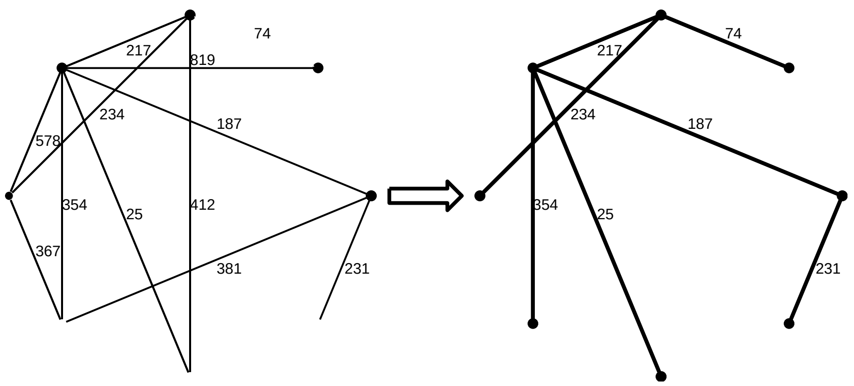
Prac 2

Name :	Student Number:
Name :	Student Number:
Mark awarded : /65	Demonstrator
I declare that this assignment is entirely my own work:	

Instructions
This practical is longer than last week's and you are *encouraged* to do the work in pairs. You and your prac partner must have the same tutor. Please hand in only one submission per pair. You are free to divide the work as you see fit. As this is the last practical you are to required to submit it in the CS2 submissions box by *5:00pm on Saturday 27 May*.

Random Graphs
Example code is provided which will generate random, weighted, directed graphs. These graphs are available in a matrix format, or as a list of edges. In addition it is possible to specify a seed number so that the same random graph is always generated. You will find the code useful for the first two questions.

1 Minimal Spanning Tree.



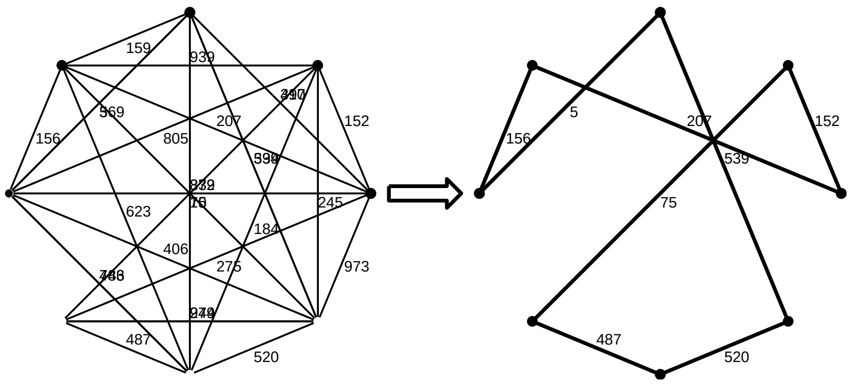
(a) (12 marks) Implement the algorithm given in the notes to find the minimal spanning tree[1].

(b) (2 marks) Provide timing information for undirected graphs of size 10,100,500,1000.

(c) (1 mark) What is the sum of all the edges forming the minimal spanning tree of the graph with 10 vertices and a seed number of 1?

[1]The minimal spanning tree as described in the notes only makes sense for undirected graphs, so remember to call `Graph.undirected()` in the given code.

2 The Traveling Salesman.

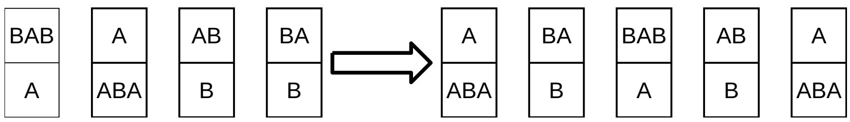


(a) (12 marks) Write a program which finds the shortest valid path that visits every node exactly once and returns to the initial starting vertex.

(b) (2 marks) Provide timing information for random graphs of sizes 10,12,14,16,18 and 20.

(c) (1 mark) How long is the shortest path for the graph with 10 vertices and a seed number of 1?

3 Post's Correspondence problem.



On the course website you will find nine instances of Post's Correspondence problem. There is also some skeleton code which you might find useful.

(a) (12 marks) Fill in the `solve()` method. Since the problem is undecidable you will need to limit your search in some way. You should also test that the solution is potentially solvable.

(b) (3 marks) Describe how you bounded your search, and how you tested the impossibility of the solutions.

(c) (5 marks) Provide a classification of each of the nine instances as either solvable (Giving the length of the solution), unsolvable or unknown. Your classification will be marked as follows:

- No marks for problems classified as unknown.
- +1 for every correct classification.
- -1 for every incorrect classification.

4 General questions

(To be answered on paper)

(a) (5 marks) Design a Turing Machine which reverses a binary string. If this Turing machine is started with an input tape of '100111' then your machine should halt with the input tape reading '111001'.

(b) (5 marks) In lectures we covered a Turing Machine which was able to count down to zero. Now design a machine which starts with a blank tape and counts up (i.e. this machine doesn't halt).

(c) (5 marks) Design a Turing Machine which converts from unary to binary. i.e. given a string '111111' (6 ones) your machine should be able to halt with output '110' (6 in binary) left on the tape. (Hint: The machine from the previous question should be useful.)