MAT1830 - Discrete Mathematics for Computer Science
Tutorial Sheet #11 and Additional Practice Questions

Tutorial Questions

1. (a) Draw the simple graph with adjacency matrix

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

using V_1, V_2, V_3, V_4 as the names for the vertices corresponding to columns 1, 2, 3, 4 respectively.

(b) Find the number of walks of length 3 from V_1 to V_2 in the graph.

(c) Without any calculation show that the top row of M^n for any even $n \geq 2$ is "1 0 0 0".

2. For each of the graphs described below, either give an example of such a graph or explain why it does not exist.

(a) A simple graph with 7 vertices with degrees 8, 5, 4, 4, 3, 1, 1.

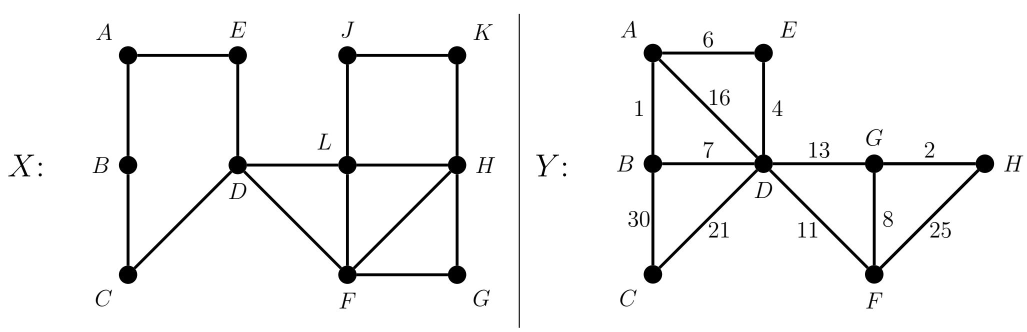
(b) A simple graph with 8 vertices with degrees 4, 4, 4, 4, 4, 2, 2, 2.

(c) A simple graph with 6 vertices with degrees 3, 3, 2, 2, 1, 1 that is a tree.

(d) A simple graph with 8 vertices with degrees 3, 3, 2, 2, 1, 1, 1, 1 that is a tree.

(e) A simple graph with 6 vertices with degrees 4, 4, 3, 3, 2, 1.

3.



(a) Find a spanning tree of X that contains neither the edge JL nor the edge KH, or explain why one does not exist.

(b) Find a spanning tree of X that contains neither the edge DL nor the edge KH, or explain why one does not exist.

(c) How many different spanning trees of X are there that contain no edge in $\{DE, FH, FL, HL\}$?

4. (a) Apply the algorithm given in lectures to find an Euler trail in the graph X.

(b) Apply the algorithm given in lectures to find a minimum weight spanning tree in the weighted graph marked Y.

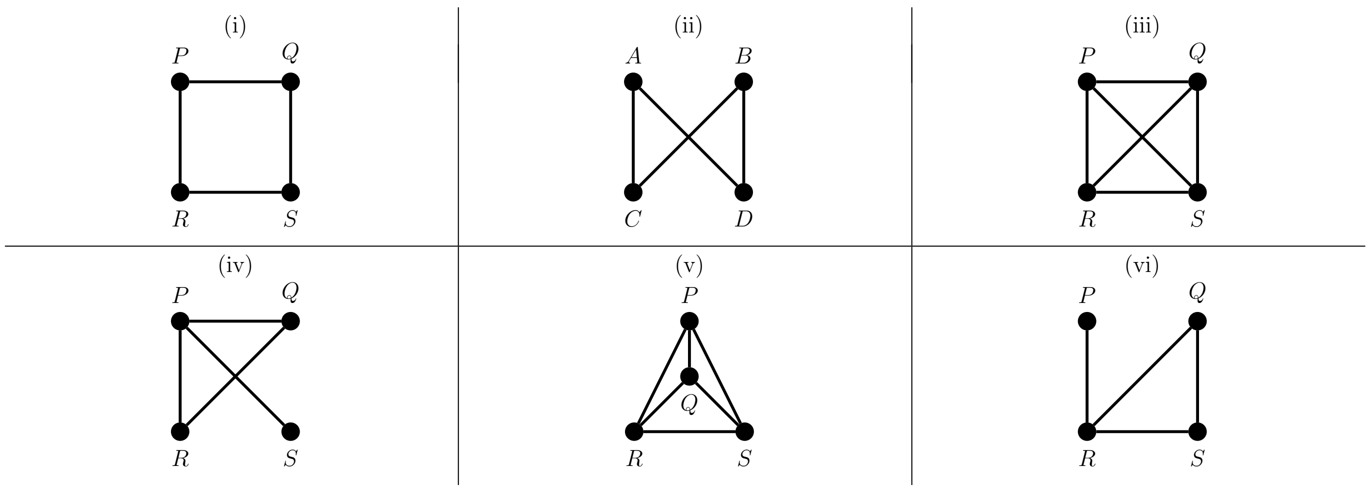
5. (a) Does every simple graph with 8 vertices and 11 edges have a spanning tree? If so, prove it. If not, give an example of a graph with 8 vertices and 11 edges with no spanning tree.

(b) If a simple graph with 8 vertices and 11 edges does have a spanning tree, how many edges will the spanning tree have?

(See over for practice questions.)

Practice Questions

1. Two graphs are equal if they have exactly the same lists of vertices and edges. They are isomorphic if we can "rename" the vertices of one graph to make it equal to the other. Which of the following graphs are equal? Which are isomorphic? How would you prove this?



2. How would you change the definition of isomorphic graphs given above to make it more formal?

3. (a) What would you do if you saw someone drive past in a car that was isomorphic to yours?
   (b) What would you do if you saw someone drive past in a car that was equal to yours?

4. A new phone company has built exchanges in Blackburn, Clayton, Frankston, Thornbury, Sunshine and Richmond and wants to connect them up with fibre optic cable as cheaply as possible. They do not care how the exchanges are connected provided that every exchange can be reached from every other exchange (possibly via other exchanges). They have done a cost analysis for laying cable between pairs of exchanges and the results are as follows.

	Blackburn	Clayton	Frankston	Thornbury	Sunshine	Richmond
Blackburn	-	$1.2M	$3.3M	$2.4M	$3.0M	$1.1M
Clayton	-	-	$2.8M	$2.1M	$3.1M	$1.3M
Frankston	-	-	-	$4.1M	$5.1M	$3.2M
Thornbury	-	-	-	-	$3.1M	$1.5M
Sunshine	-	-	-	-	-	$2.7M
Richmond	-	-	-	-	-	-

   How cheaply can they do it?

5. There are 21 people at a party. Some of them have met before and others have not. Omar offers to buy everyone at the party dinner if it turns out that everyone there has previously met an odd number of the other guests. Is he taking a risk?

6. Is it always possible to find a closed walk in a simple connected graph G that uses every edge exactly twice? Explain why or why not.

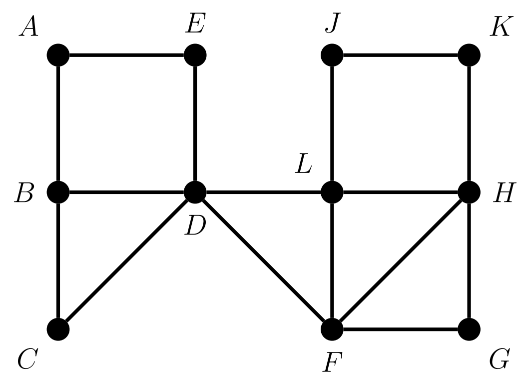
Some questions focussing on the lecture 33 material are on the next page of the electronic version of this sheet (see moodle).

Practice questions for lecture 33

1. (a) Use the breadth first algorithm given in lectures to find a spanning tree in the graph below.

   (b) Use the depth first algorithm given in lectures to find a spanning tree in the graph below.

   [**Note:** In (a) and (b), where the algorithms allow a choice of which vertex to add to a queue or stack, choose the vertex earliest in the alphabet. In particular, A should be chosen as the root in both (a) and (b).]



2. Suppose you are given a connected graph G and you wish to find a path between two vertices A and B in G

   (a) How could you use a spanning tree (or part of one) to accomplish this.

   (b) Suppose you wanted to find a path that was definitely one of the shortest paths between A and B. Would you use a depth-first or breadth-first algorithm to find the spanning tree? Why?