This is Assignment 9 for MATH1005 students in a Friday workshop. It is due at 6 pm on the Thursday after Workshop 9 (6 days after was released).

There are four problems. The numbering of the problems is strange because the numbering is taken from a much larger document that has many problems from which I can select. As long as you can see four different problems, then you have the complete assignment.

You should write your best solutions to the problems here, and then upload your solutions before the due time. Here are three ways you may complete the assignment:

1. Print the assignment sheet. Write your solutions in pen or pencil on the print out. Scan your completed assignment, turn the file into a single .pdf file, then upload your solution file to Wattle.

2. Write your solutions in pen or pencil on blank paper. You should clearly label your solutions and you should write them in the order in which the problems appear in your assignment. Scan your completed assignment, turn the file into a single .pdf file, then upload your solution file to Wattle.

3. Download the assignment sheet to a tablet. Annotate the file using your favourite annotation software. **Flatten the file**–this makes your annotations a permanent part of the file, and if you do not do this then we see only a blank assignment in our grading software. Upload your flattened solution file to Wattle.

In all cases, the file you upload must be a .pdf file.

Please remember to plan your time carefully so you are not trying to submit your assignment at the last minute. No late work is accepted.

Please enjoy,

AP

Question 1† Circle the correct answers.

For this question, working is not required and will not be marked.

Here is a list of some properties that a graph G may, or may not, possess:

$p(G)$**:** G is connected

$q(G)$**:** Every edge of G is a bridge

$r(G)$**:** Any two vertices of G are connected by exactly one simple path

$s(G)$**:** G is simple

$t(G)$**:** G is a tree

$u(G)$**:** G contains a vertex of degree 1

$v(G)$**:** $|V(G)| = |E(G)| + 1$

$w(G)$**:** G contains no simple circuits

Decide the truth or falsity of each of the implication statements below. Circle accordingly.

(a) For all finite graphs G $\quad p(G) \wedge q(G) \Rightarrow r(G)$ TRUE / FALSE

(b) For all finite graphs G $\quad q(G) \wedge r(G) \Rightarrow s(G)$ TRUE / FALSE

(c) For all finite graphs G $\quad r(G) \wedge s(G) \Rightarrow t(G)$ TRUE / FALSE

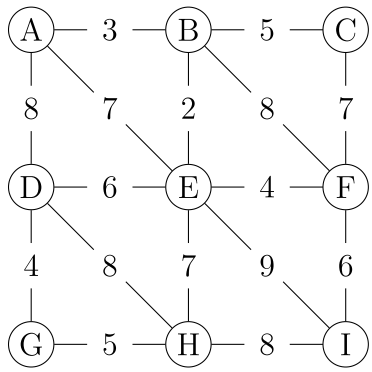
(d) For all finite graphs G $\quad s(G) \wedge t(G) \Rightarrow u(G)$ TRUE / FALSE

(e) For all finite graphs G $\quad t(G) \wedge u(G) \Rightarrow v(G)$ TRUE / FALSE

(f) For all finite graphs G $\quad u(G) \wedge v(G) \Rightarrow w(G)$ TRUE / FALSE

Question 3[†]

Write the correct values in the boxes.



For this question, working is not required and will not be marked.

A diagram for a weighted graph G is shown at right.

On a separate piece of paper, not to be submitted, construct a minimal spanning tree for G, so that you can answer parts (a) - (c) below. As it happens, there is only one minimal spanning tree.

(a) In G, vertex E has degree 6.
But in the spanning tree its degree is $\square$.

(b) The total weight of the spanning tree is $\square$.

(c) Using only the spanning tree, the distance from
A to I (*i.e* the total weight of the path from A to I) is $\square$.

Now, again on a separate peice of paper, not to be submitted, use Dijkstra's algorithm to find the shortest path from A to I. (Of course, on a tiny graph like this you can easily find the shortest path 'by eye', but that's not the point. The point is to check your understanding of the algorithm.)
Use your work to answer parts (d) - (f) below.

(d) The shortest distance from A to I on G
(i.e. the total weight of shortest path from A to I) is $\square$.

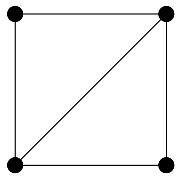
(e) Dijkstra's algorithm creates a tree comprising
all the locked-in vertices and edges.
The number of vertices on this 'Dijkstra tree' is $\square$.

(f) The total weight of the Dijkstra tree is $\square$.

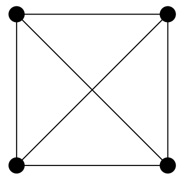
Question 4†

Write the correct integer values in the boxes.

For this question, working is not required and will not be marked.

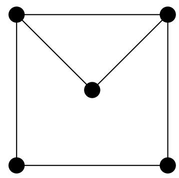


This question is about the number of spanning trees of a graph. In a lecture we used complementary counting to calculate that the graph depicted at left has exactly eight spanning trees. By adding just one more edge to this graph we arrive at the complete graph K_4 depicted at right.



A spanning tree has $n-1 = 3$ edges and so we first count the total number of sets of three edges; our extra edge takes this from $\binom{5}{3}$ to $\binom{6}{3}$. Now we count the number of sets to discard because they do not create a tree. The extra edge means there are a couple more of these. This leads to:

(a) The number of spanning trees of K_4 is $\square$.

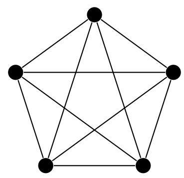


Moving up to $n = 5$ vertices, we next consider the graph G depicted at right, which has five vertices and six edges. Again using complementary counting we need to find two numbers:

(b) The total number of sets of edges we need to consider is $\square$ [number, not formula]

(c) The number of these sets that do not provide a tree is $\square$.

By subtracting answer (c) from answer (b) you will get the number of spanning trees for G. You can check your answers by drawing and counting all these trees.



Finally consider the case of the complete graph K_5 depicted at right. It again has five vertices, but now ten edges, so there will be a lot more spanning trees. Counting as before:

(d) The total number of sets of edges we need to consider is $\square$ [number, not formula]

(e) The number of these sets which need to be discarded because they form **disconnected** subgraphs (and hence cannot be trees) is $\square$.

(f) The number of these sets which need to be discarded because they form **connected but not circuit-free** subgraphs (and hence cannot be trees) is $\square$. Hint: Two types of circuits are involved. There are a lot.

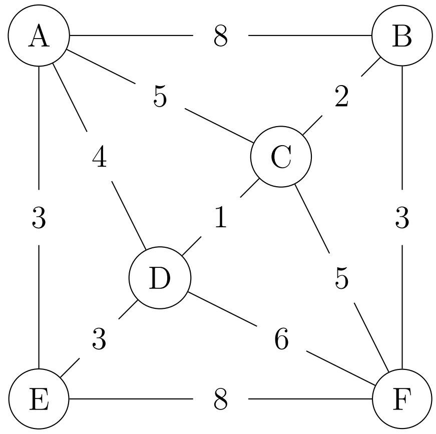
By subtracting answers (e) and (f) from answer (d) you will get the number of spanning trees for K_5. You might spot a pattern by comparing the corresponding values for K_4 and K_3. The simple formula that applies, and especially the proof of that formula, is one of the jewels of graph theory. Try Googling it.

Question 6† Circle the correct answers.

For this question, working is not required and will not be marked.

Working on a separate piece of paper, not to be submitted, apply Dijkstra's shortest path algorithm to the weighted graph depicted at right, with the objective of finding a shortest path from A to F. As you work through the algorithm, circle the correct completions to the statements below.

This question is solely about the use of Dijkstra's algorithm, so you must strictly adhere to it, and not stray to any variations of your own.



(a) The first two vertices to be locked in are A and E.
The next vertex to be locked in is B / C / D / F

(b) The number of vertices that get labelled and then re-labelled during the application of the algorithm is 0 / 1 / 2 / 3

(c) The first label to be applied to F is B10 / B11 / C10 / D10 / E11

(d) The particular shortest path from A to F found by the algorithm is
ACBF / ACF / ADCBF / ADCF / ADF

(e) The tree of locked edges and vertices that results from applying the algorithm is
a minimal spanning tree / a non-minmal spanning tree / not a spanning tree

(f) Suppose that Dijkstra's algorithm is applied to the same weighted graph, but this time to find a shortest path in the reverse direction, from F to A.
The path found is FBCA / FCA / FBCDA / FCDA / FDA