Question 1

(a) Explain why every finite tree of order 2 or more has at least one vertex of degree 1.

Since there are at least two vertices, and every tree is connected, there can be no vertices of degree 0. *So every vertex has at least one adjacent vertex. (Trees have no loops.)*

Pick any vertex v. *If it is not of degree* 1 *move to an adjacent vertex* w. *If* w *does not have degree* 1 *then it has degree* 2 *or more and so has an adjacent vertex* $x \neq w$. *Move to* x. *If* x *does not have degree 1 then there exists an adjacent vertex* y *such that* $y \neq w$. *Continue this process, never moving back along an edge you just followed.*

Since the tree is finite, the process must eventually stop at a vertex of degree 1*, otherwise you would revisit a vertex visited earlier and hence would have traversed a circuit. But trees do not have circuits.*

(b) Prove by mathematical induction on n that every tree of order $n \geq 1$ has exactly $(n-1)$ edges. (You will need part (a).)

Let $P(n)$*: Every tree of order* $n \geq 1$ *has exactly* $(n-1)$ *edges.*

Basis step: *For* $n = 1$ *there are* 0 $(= n-1)$ *edges since the only other possibility would be loops, but trees don't have loops. Hence* $P(1)$ *holds.*

Inductive step: *Let* $n \in \mathbb{N}$. *Suppose that* $P(1), P(2), \ldots, P(n)$ *all hold. We shall consider* $P(n+1)$. *Let* T *be a tree of order* $n+1$. *By part (a)* T *has a vertex* v *of degree* 1. *Let* e *be the (unique) edge of* T *incident on* v*, and let* S *be the subgraph of* T *obtained by removing* v *and* e *from* T. *Then*

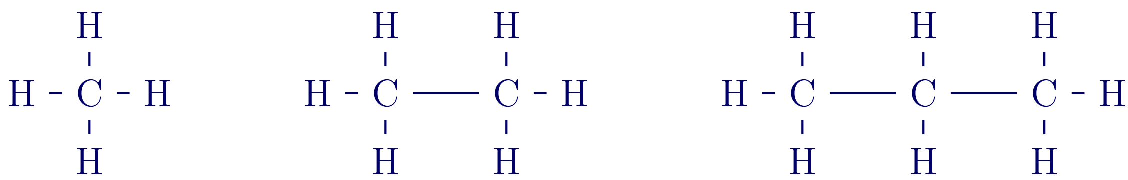
- S *has circuits, since* T *has none.*
- S *is connected, since* T *is connected and removal of* v *and* e *does not disconnect it because, with degree* 1, v*, and therefore also* e*, cannot be part of any path connecting two other vertices.*

So S *is a tree with* n *vertices (one less than* T*). By* $P(n)$, S *has* $n-1$ *edges. Since* S *has one less edge than* T, T *has* $(n{-}1)+1 = n = (n{+}1)-1$ *edges. Hence* $P(n+1)$ *holds.*

By the Principle of Mathematical Induction, $P(n)$ *for all* $n \in \mathbb{N}$.

Question 2 Recall that a hydrocarbon molecule consists of joined-up carbon and hydrogen atoms. If the molecule contains c carbon atoms and h hydrogen atoms we write its formula as $\mathrm{C}_c\mathrm{H}_h$. The way the atoms are joined up can be modelled by a connected graph with the atoms as vertices and the joins as edges. In such a graph every carbon vertex has degree 4 and every hydrogen vertex has degree 1. The graph may have parallel edges but cannot have loops. A hydrocarbon molecule is called *saturated* if it contains the maximum possible number of hydrogen atoms for its number of carbon atoms.

(a) Draw the graphs of saturated hydrocarbon molecules containing 1 (methane $\mathrm{CH_4}$), 2 (ethane $\mathrm{C_2H_6}$) and 3 (propane $\mathrm{C_3H_8}$) carbon atoms.



(b) Explain why the graph of a saturated hydrocarbon molecule must be a tree. [Hint: What is the defining property of a tree?

Molecules are connected by definition.

We need to show that a saturated hydrocarbon molecule cannot contain a circuit. Since hydrogen atoms have degree 1*, such a circuit must be an all-carbn 'ring'.*

However, breaking one link of the ring and joining an H *to each free end increases the* H*-count but does not disconnect the graph. So a molecule with a* C*-ring is not saturated.*

(c) Recall that a tree with n vertices has exactly $n-1$ edges. (See Question 1). Use this to explain why the total degree of the graph of a saturated hydrocarbon molecule is $2c+2h-2$.

By the handshake theorem, total degree is twice the number of edges.

Since $n = c + h$ this gives a total degree of $2(c+h-1) = 2c+2h-2$.

(d) Give another formula for the total degree of the graph of *any* hydrocarbon molecule.

Since $deg(\mathrm{C}) = 4$ and $deg(\mathrm{H}) = 1$,

$$\textit{total degree} = c \times 4 \;+\; h \times 1 = 4c + h.$$

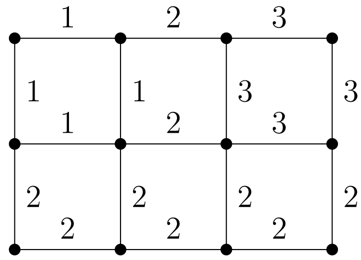
(e) Use (c) and (d) to prove Cayley's result that every saturated hydrocarbon molecule has formula $\mathrm{C}_c\mathrm{H}_{2c+2}$.

Equating the two formulas for total degree gives

$$2c + 2h - 2 = 4c + h$$

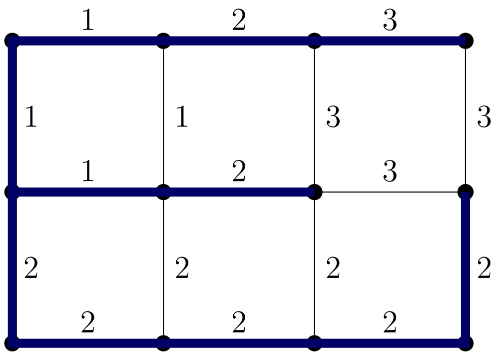
leading to $h = 2c + 2$. So $\mathrm{C}_c\mathrm{H}_h = \mathrm{C}_c\mathrm{H}_{2c+2}$.

Question 3 A weighted graph G is shown at right.



(a) Find a minimal spanning tree for G and calculate its total weight.

Trace out your answer below:



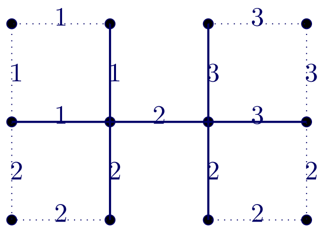
This is just one of many possible answers.

Total weight
$= 3\times 1 \;+\; 7\times 2 \;+\; 1\times 3$
$= 3 + 14 + 3 = \boxed{20}$

(b) Prove or disprove that G has a minimal spanning tree containing two vertices of degree 4.

Only the two central vertices of G have degree 4 and there are no vertices of higher degree.

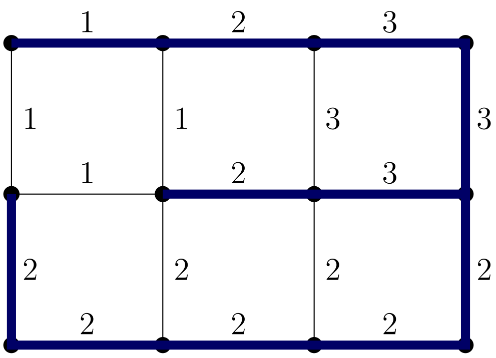
So a spanning tree with two vertices of degree 4 must include these two central vertices and all the edges incident on them. This contributes $1{+}1{+}2{+}2{+}2{+}3{+}3 = 14$ *to the total weight.*



Then to include the corner vertices, no matter which joining edges we choose, we must add another $1{+}3{+}2{+}2 = 8$ *to the total weight.*

Thus the total weight is $14{+}8 = 22 > 20$, *and so the tree is not minimal.*

(c) Find a maximal spanning tree for G and calculate its total weight.
Trace out your answer below:



Using the minimimal spanning tree algorithm (Kruskal's), but picking maximal available edges instead of minimal ones, gives a maximal spanning tree. One of many is shown.

Total weight $= 3\times 3 \;+\; 7\times 2 \;+\; 1\times 1$
$= 9 + 14 + 1 = \boxed{24}$.

Question 4 An irrigation system is being laid down to connect 7 fields. The cost of the pipelines between field i and field j is given by the matrix at right.

Find a set of pipelines that connects all fields at a minimal cost. Branching is allowed.

$$\begin{array}{cc} & \begin{matrix} A & B & C & D & E & F & G \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \\ G \end{matrix} & \begin{pmatrix} 0 & 1 & 4 & 5 & 3 & 6 & 1 \\ 1 & 0 & 1 & 8 & 2 & 3 & 7 \\ 4 & 1 & 0 & 6 & 1 & 3 & 5 \\ 5 & 8 & 6 & 0 & 6 & 1 & 4 \\ 3 & 2 & 1 & 6 & 0 & 8 & 9 \\ 6 & 3 & 3 & 1 & 8 & 0 & 4 \\ 1 & 7 & 5 & 4 & 9 & 4 & 0 \end{pmatrix} \end{array}$$

The given information determines a complete weighted graph with fields as vertices and edges weighted by cost. We seek a minimal spanning tree for this weighted graph.

There is no need to draw up the whole weighted graph.

We can apply Kruskal's algorithm directly to the matrix, each time picking the lowest number that hasn't been picked already and whose corresponding edge does not create a circuit.

The matrix is symmetric, so we can ignore all entries below the diagonal of zeros.

Pick pipelines in this order:

Cost 1: AB, AG, BC, CE, DF (none create circuits)
Cost 2: none (BE would create a circuit)
Cost 3: BF (AE and CF would create circuits,
or we could have chosen CF, then
AE and BF would have created circuits)

At this stage we have added 6 edges. Since there are only 7 vertices, we are done. The resulting minimal spanning tree can be drawn in many ways, since we don't know locations of the fields. One way for each of the two possibilities (for choice of cost 3 pipeline) is shown below.

$$\boxed{\begin{tikzcd} F \arrow[r, dash] \arrow[d, dash] & B \arrow[r, dash] \arrow[d, dash] & A \arrow[r, dash] & G \\ D & C \arrow[r, dash] & E & \end{tikzcd}} \qquad \textit{OR} \qquad \boxed{\begin{tikzcd} D \arrow[r, dash] & F \arrow[r, dash] \arrow[d, dash] & B \arrow[r, dash] & A \arrow[d, dash] \\ & C \arrow[r, dash] & E & G \end{tikzcd}}$$

The total cost of the pipelines is $1+1+1+1+1+3=8$.
(This was not asked for.)

Question 5 To see how bad the (greedy) Nearest Neighbour algorithm can be consider this deliberately pathological example: The complete graph K_{10} has vertices labelled 1,...,10 and each edge $\{a, b\}$ is given weight ab. Do not attempt to draw this graph.

(a) Starting at the vertex determined by the last digit of your ANU ID (use vertex 10 if that digit is 0) , apply the Nearest Neighbour algorithm to find a Hamilton circuit of supposedly low total weight. Give your answer as a list of vertices, and calculate the total weight.

Example: The last digit of my ID is 8. By Nearest Neighbour I next visit b where $8b$ is as small as possible. So $b = 1$. Continuing in this manner, and avoiding already visited vertices, gives

$$\begin{array}{ll} \textit{vertices:} & 8-1-2-3-4-5-6-7-9-10-8 \\ \textit{weights:} & 8+2+6+12+20+30+42+63+90+80=353. \end{array}$$

For checking, other totals are:

Start:	1	2	3	4	5	6	7	8	9	10
Total:	340	347	353	357	359	359	357	353	347	340

(b) By avoiding large products such as 9×10, find (without the help of any algorithm) a Hamilton circuit of much lower total weight. Again give your answer as a list of vertices, and calculate the total weight.

Try alternating high and low values:

$$\begin{array}{ll} \textit{vertices:} & 1-10-2-9-3-8-4-7-5-6-1 \\ \textit{weights:} & 10+20+18+21+24+32+28+35+30+6=230. \end{array}$$

Can you do better? Yes you can, but the point is that this is already **much** *better than Nearest Neighbour gives, no matter where you start.*

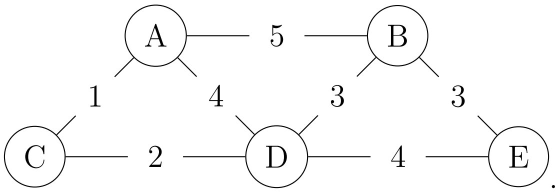
(c) How many different Hamilton circuits would you need to consider in order to prove your answer to (b) has the lowest possible total weight? Circuits are different if and only if there is at least one edge on one that is not on the other.

In a circuit, the start point is irrelevant. There are 9 choices for the next vertex, 8 choices for the third vertex and so on, giving 9! circuits. But each circuit will be counted in both directions, so we get

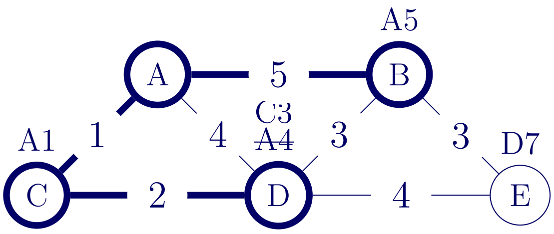
$$\frac{9!}{2} = \boxed{181\,440} \textit{ Hamilton circuits.}$$

That's a lot to check!

Question 6 Let G be the weighted graph



(a) Show that Dijkstra's algorithm correctly finds that the shortest path from A to B is the direct path using just the edge AB. You can do all the work on a single diagram, but, to show that you have used the algorithm correctly, if an annotation needs updating do not erase it — just put a line through it and write the new annotation above that.



(b) In what order are the vertices added to the tree?

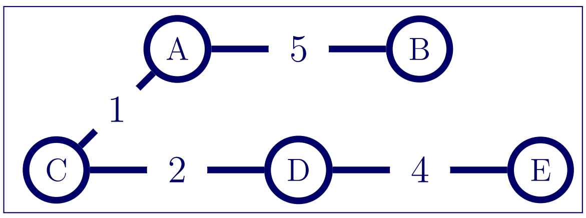
(A), C, D, B

(c) Notice that the algorithm does not, in this instance, generate a spanning tree. Which vertex or vertices are missing?

E

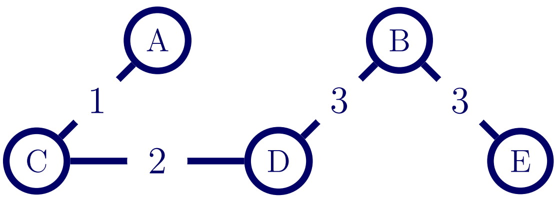
(d) Extend the use of the algorithm until the shortest distance from A to each other vertex is established, and a spanning tree is thereby generated. Draw this tree.

Vertex E is marked with D, so add edge DE to give the spanning tree:



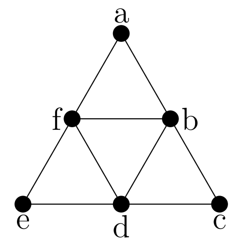
(e) Is the spanning tree you have generated using Dijkstra's algorithm a minimal spanning tree? Justify your answer.

No. *Kruskal's algorithm gives*



which has total weight $1+2+3+3=9$. *This total is less than that for the Dijkstra tree, which has total weight* $1+2+4+5=12$.

Question 7 For the graph H at right:



(a) Is the graph simple? Justify your answer.

(b) Is the walk abcdb

(i) a path?
(ii) closed?
(iii) simple?

(c) Find simple circuits of lengths 3, 4 and 5.

(d) Find an Euler circuit.

(e) Find a Hamilton circuit.

(a) Yes, the graph is simple. There are no loops and no parallel edges.

(b) (i) The walk is a path because no edge is used twice.

(ii) The walk is not closed because the initial and terminal vertices are different.

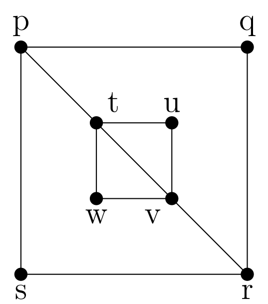
(iii) The walk is not simple because there is a repeated vertex (the vertex b is visited twice), which is only allowed in simple walks if the walk is closed and the repeated vertex is the initial and terminal vertex.

(c) For example: $abfa$ is a simple circuit of length 3; $bcdfb$ is a simple circuit of length 4; $bcdefb$ is a simple circuit of length 5.

(d) (We note that the graph is connected and every vertex has even degree, so there is an Euler circuit). For example, the following is an Euler circuit: $abcdefbdfa$.

(e) For example, the following is a Hamilton circuit: $abcdefa$.

Question 8 For the graph J at right:



(a) Prove that J has no Euler circuit.

(b) Prove that J has no Hamilton circuit.

(c) Suppose that Fleury's algorithm is used to find an Euler path from p to r. What feature of the algorithm prevents the path starting ptvr?

1. *We have a theorem that says: A connected graph has an Euler circuit if and only if every vertex has even degree. Since p has degree 3, J has no Euler circuit.*

2. **We shall use a proof by contradiction. Suppose that J has a Hamilton circuit.** *Then J has an Euler circuit that starts and ends at w. Since J has 8 vertices, the Hamilton circuit has 9 vertices; the first and the ninth vertices are w. Since the only neighbours of w are t and v, the second vertex in the circuit must be t or v and the second last vertex in the circuit must be t or v. Since t and v are each visited once by the circuit, either t is the second vertex visited and v is the eighth vertex visited, or v is the second vertex visited and t is the eighth vertex visited. In the latter case, we may reverse the circuit and we still have a Hamilton circuit. So, without loss of generality, we may assume that t is the second vertex visited and v is the eighth vertex visited; that is, the circuit reads $wt \dots vw$.*

   The only neighbours of u are t and v. It follows that the circuit must visit u immediately after visiting one of t or v. Since, in this case, the circuit returns to w immediately after visiting v, we have that the circuit must visit u immediately after visiting t. That is, the circuit reads $wtu \dots vw$. Since t is the second vertex and u is the third vertex in the circuit, and the only neighbours of u are t and v, it must be that v is the fourth vertex in the circuit. So the circuit reads $wtuv \dots vw$. But then v is the fourth vertex and the eighth vertex visited in the circuit. This contradicts the fact that the circuit is a Hamilton circuit.

   Since our supposition that J has a Hamilton circuit has led to a contradiction, it cannot hold. $\square$

3. *If Fleury's algorithm began by constructing a path ptv, then the edge $\{v, r\}$ is a bridge in the remaining graph and would not be chosen as the next edge.*