5.4

CONTOUREDUCATION

Website: contoureducation.com.au | **Phone:** 1800 888 300
Email: hello@contoureducation.com.au

VCE Specialist Mathematics ½
Graph Theory II [5.4]
Workbook

Outline:

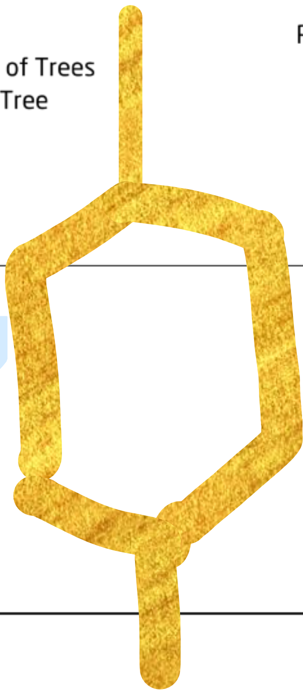
Recap of Graph Theory I Pg 2-4

Walk of a Graph (12D) Pg 5-14
- Walk of a Graph
- Length of a Walk
- Counting the Number of Walks

Euler Trail and Circuits (12B) Pg 15-28
- Definition of Euler Trails
- Identifying Whether a Graph has an Euler Trail
- Finding Euler Trails Efficiently
- Definition of Euler Circuit
- Identifying Whether a Graph has an Euler Circuit

Hamiltonian Paths and Cycles (12C) Pg 29-33
- Definition of Hamiltonian Paths
- Definition of Hamiltonian Cycles

Trees Pg 34-44
- Definition of Trees
- Spanning Tree



Complex no.

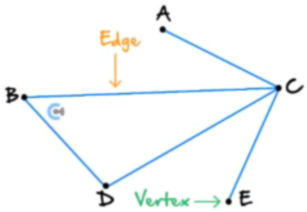
Vectors

Section A: Recap of Graph Theory I

Cheat Sheet

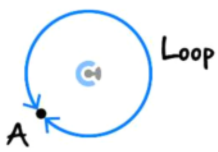
[5.3.1] – Graph theory fundamentals – vertices, edges, degree, adjacency lists and matrices

- **Vertices and Edges**



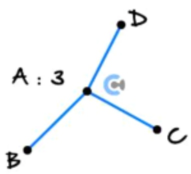
- A graph consists of a set of points called vertices and a set of unordered pairs of vertices, called edges.

- **Loops**



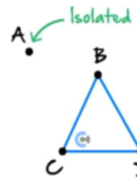
- Loop is an edge which connects to the same vertex.

- **Degree of a Vertex**



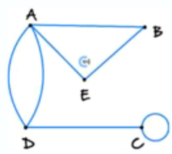
- Degree of a vertex is the number of ______________ connected to the vertex.

- **Isolated Vertex**



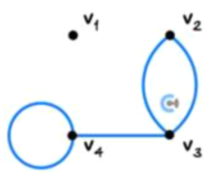
- Isolated vertex has no edges connected to it.
- Its degree is equal to __________.

- **Adjacency Lists**

Graph	Adjacency List
	$A \to (B, D, D, E)$ $B \to (A, E)$ $C \to (C, D)$ $D \to (A, A, C)$ $E \to (A, B)$

- Adjacency list contains all the vertices a given vertex is connected to.
- If the point is connected multiple times, we write the vertex multiple times.
- If a point is looped with itself, we write the vertex to be adjacent to itself.

- **Adjacency Matrix**



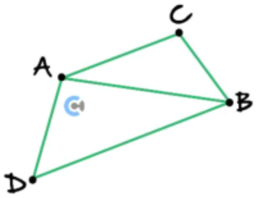
$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

- A matrix that represents the vertices and edges that connect the vertices of a graph.

Cheat Sheet

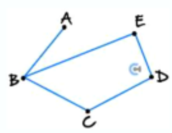
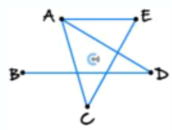
[5.3.2] – Types of graphs

Simple Graph



- A simple graph is one in which pairs of vertices are connected by ______________________________.

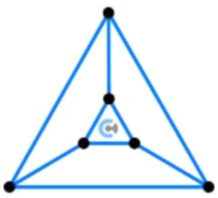
The Complement of a Simple Graph

Simple Graph	The Complement
	

Complement of $G = \overline{G}$

- The complement of a simple graph contains the same set of vertices.
- But it contains a complementary set of edges. (Edges that the original graph does not have.)

Regular Graphs



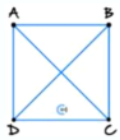
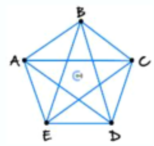
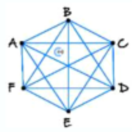
- Regular graph has all its vertices with the same ____________.
- If each vertex has a degree r then the graph is "regular of degree r" or "r-regular".

Number of Edges and Degree of All Vertices of a Regular Graph

$$\textbf{Number of Edges} \times 2 = \textbf{Total Degree of all Vertices}$$

- The above result is commonly known as the Handshake Lemma.

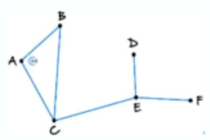
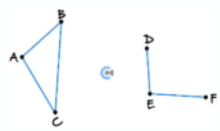
Complete Graph (K_n)

K_4	K_5	K_6
		

- A complete graph is a simple graph in which each vertex is connected to every other vertex.
- A complete graph is denoted by K_n, where n is the number of vertices in the graph.
- A complete graph is a type of regular graph.
- For K_n:

$$\textbf{Number of Edges for Complete Graph} = \_\_\_\_\_\_\_\_\_\_$$

Connected Graphs

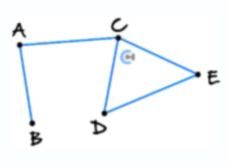
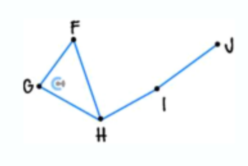
Connected Graph	Disconnected Graph
	

- A connected graph is a graph where it is possible to reach all vertices by moving along edges.
- A graph that is not connected is called a disconnected graph.

Cheat Sheet

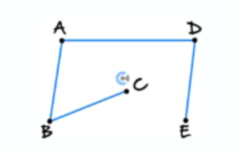
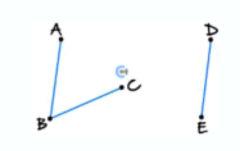
[5.3.3] – Isomorphisms and subgraphs

- **Isomorphism**

<table>
<tr><th colspan="2">Isomorphic Graph</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

  - Two graphs are **isomorphic** if their vertices and edges differ only by how they are named.
  - **Checklist for Determining Isomorphism**
    - Are the number of vertices the same in each graph?
    - Are the numbers of edges the same in each graph?
    - Check that the degrees of each vertex match for both graphs.
    - Label each vertex on both graphs and check if there is a correspondence between the vertices.

- **Subgraphs**

<table>
<tr><th>Original Graph</th><th>Subgraph</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

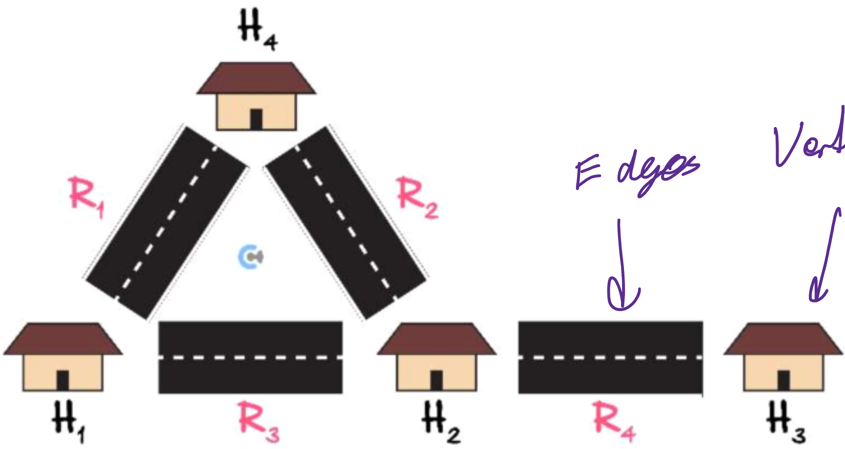
  - A subgraph is a graph whose vertices and edges are all contained within the original graph.
  - A subgraph can be created by removing edges and vertices from the original graph.

Section B: Walk of a Graph (12D)

Sub-Section: Walk of a Graph

Analogy: Walk of a Graph

- Krish goes around the Contour village.



Edges Vertices

- He starts from H_1 and visits H_2 then H_4.

How can we describe Krish's path?

- We can describe his path with the houses he visited and the road he took.

path

$$H_1, R_3, H_2, R_2, H_4$$

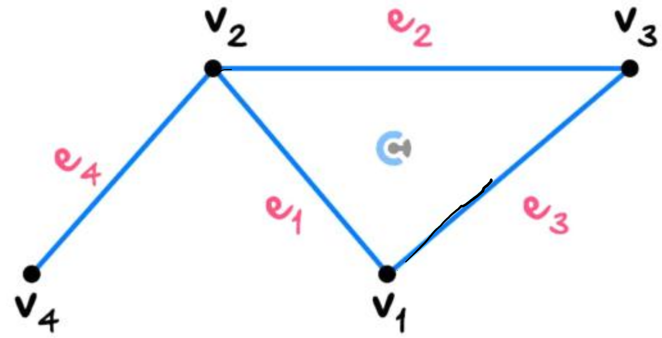
Walk of a Graph

$$v_1, e_1, \cdots, v_{n-1}, e_1, v_n$$

- Order of the vertices (points) and edges (lines) we are visiting.
- Alternates between a vertex and an edge.
- If there is only one edge between two vertices, stating an edge can be abbreviated.

Question 1

Consider the graph below:



State the walk that visits all vertices once, starting with v_1 and ending with v_4. (Vertices do not have to be in an ascending order.)

$V_1, e_3, V_3, e_2, V_2, e_4, V_4$

V_1, V_3, V_2, V_4

<u>Discussion:</u> Did we need to write the edges for the previous question?

Not

Space for Personal Notes

Sub-Section: Length of a Walk

Discussion: How many edges did we have to go through for the previous question?

3 ← length of a walk

This is called the length of a walk!

Length of a Walk

$$\textbf{Length of a Walk} = \textbf{Number of Edges in a Walk}$$

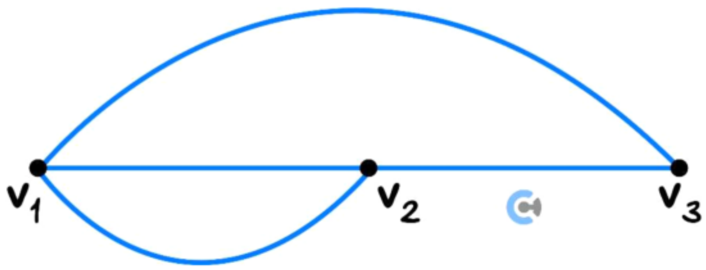
- Number of edges in a walk.
- If we use the same edge multiple times, we count it multiple times for the length.

Space for Personal Notes

Sub-Section: Counting the Number of Walks

Question 2 Walkthrough.

Consider a graph and its adjacency matrix.



Adjacency matrix $A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \end{array}$

a. State the number of walks of length 1 from v_1 to v_2.

2

b. State the element of the adjacency matrix: $A_{1,2}$.

2

c. Hence, what do you notice?

Same

Space for Personal Notes

Definition

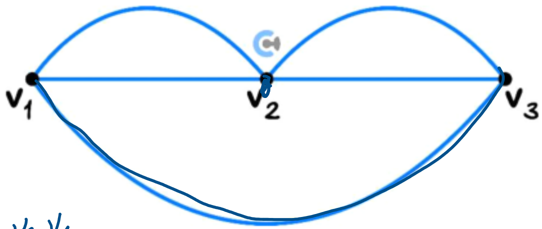
Number of Walks with the Length of 1

$$A = \begin{bmatrix} v_1, v_1 & v_1, v_2 & v_1, v_3 \\ v_1, v_2 & v_2, v_2 & v_2, v_3 \\ v_1, v_3 & v_3, v_2 & v_3, v_3 \end{bmatrix}$$

- The element gives the number of walks of length 1 between two vertices.

Question 3

Consider a graph and its adjacency matrix:



$$\text{Adjacency matrix } A = \begin{array}{c c} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} & \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix} \end{array}$$

Find the number of walks of length 1 between:

a. v_1 and v_2.

2

b. v_2 and v_3.

2

c. v_1 and v_3.

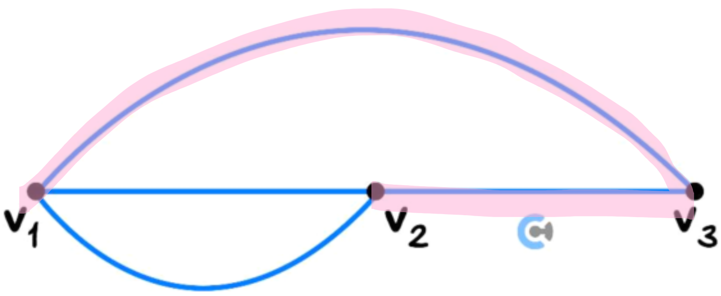
1

Space for Personal Notes

Let's now consider walks of length 2.

Question 4

Consider a graph and its adjacency matrix:



Adjacency matrix $A = \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array}\begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

a. State the number of walks of length 2 from v_1 to v_2.

1

b. Find A^2.

$A^2 =$

$$\begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 1 & 2 \\ 1 & 5 & 2 \\ 2 & 2 & 2 \end{bmatrix}$$

c. State the element of the adjacency matrix: $A^2_{1,2}$.

1

d. Hence, what do you notice?

Same as a)

Definition

<u>Number of Walks with the Length of n</u>

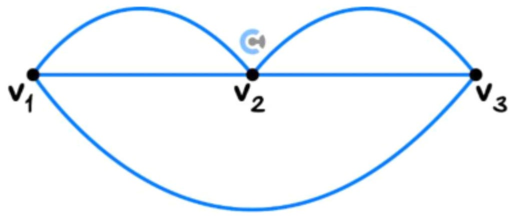
$$A^n = \begin{bmatrix} v_1, v_1 & v_1, v_2 & v_1, v_3 \\ v_1, v_2 & v_2, v_2 & v_2, v_3 \\ v_1, v_3 & v_3, v_2 & v_3, v_3 \end{bmatrix}^{n}$$

- For a walk of length n, we power the adjacency matrix by $\underline{\ n\ }$.
- The element gives the number of walks of length n between two vertices.

Space for Personal Notes

Question 5

Consider a graph and its adjacency matrix:



$$\text{Adjacency matrix } A = \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}$$

a. Find A^2.

$$A^2 = \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix} \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}$$

$$A^2 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 5 & 2 & 4 \\ 2 & 8 & 2 \\ 4 & 2 & 5 \end{bmatrix} \end{array}$$

b. Find the number of walks of length 2 between v_1 and itself.

5

c. Find the number of walks of length 2 between v_1 and v_2.

2

d. Find the number of walks of length 2 between v_2 and v_3.

2

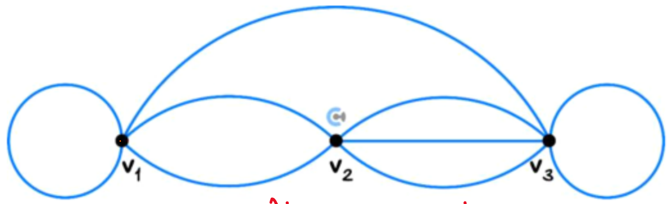
e. Find the number of walks of length 2 between v_1 and v_3.

4

How does this work?

Exploration: Counting the Number of Walks from the Adjacency Matrix

- Consider the following graph with the following adjacency matrix:



$$\textbf{Adjacency Matrix} = \begin{array}{c|ccc} & v_1 & v_2 & v_3 \\ \hline v_1 & n(v_1 \to v_1) & n(v_1 \to v_2) & n(v_1 \to v_3) \\ v_2 & n(v_2 \to v_1) & n(v_2 \to v_2) & n(v_2 \to v_3) \\ v_3 & n(v_3 \to v_1) & n(v_3 \to v_2) & n(v_3 \to v_3) \end{array}$$

- How many ways are there to go from v_1 to v_3 with the length of 2?

  State the possible order of the vertices we can go through.

$$v_1 \to V_1 \to v_3$$

$$v_1 \to V_2 \to v_3$$

$$v_1 \to V_3 \to v_3$$

- The number of ways to achieve each path would be a multiple of what two numbers?

$$n(v_1 \to v_1 \to v_3) = n(V_1 \to V_1) \times n(V_1 \to V_3)$$
$+$
$$n(v_1 \to v_2 \to v_3) = n(V_1 \to V_2) \times n(V_2 \to V_3)$$
$+$
$$n(v_1 \to v_3 \to v_3) = n(V_1 \to V_3) \times n(V_3 \to V_3)$$

- Now let's consider our adjacency matrix **squared**.

$$\begin{bmatrix} n(v_1 \to v_1) & n(v_1 \to v_2) & n(v_1 \to v_3) \\ n(v_2 \to v_1) & n(v_2 \to v_2) & n(v_2 \to v_3) \\ n(v_3 \to v_1) & n(v_3 \to v_2) & n(v_3 \to v_3) \end{bmatrix} \cdot \begin{bmatrix} n(v_1 \to v_1) & n(v_1 \to v_2) & n(v_1 \to v_3) \\ n(v_2 \to v_1) & n(v_2 \to v_2) & n(v_2 \to v_3) \\ n(v_3 \to v_1) & n(v_3 \to v_2) & n(v_3 \to v_3) \end{bmatrix}$$

- What do we get when we combine the first row of the first matrix (v_1) and the third column of the second matrix (v_3)?

$$\underbrace{n(v_1 \to v_1) \times n(v_1 \to v_3) + n(v_1 \to v_2) \times n(v_2 \to v_3) + n(v_1 \to v_3) \times n(v_3 \to v_3)}_{\text{Sum of length 2 walks}}$$

- What do you notice?

Sum of length 2 walks

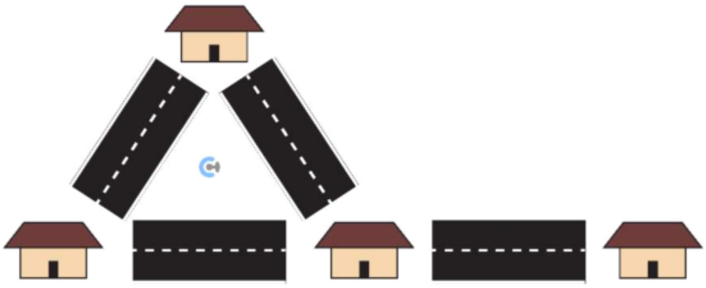
Space for Personal Notes

Section C: Euler Trail and Circuits (12B)

Sub-Section: Definition of Euler Trails

Analogy: Understanding Euler Trails

- Krish is faced with a dilemma.

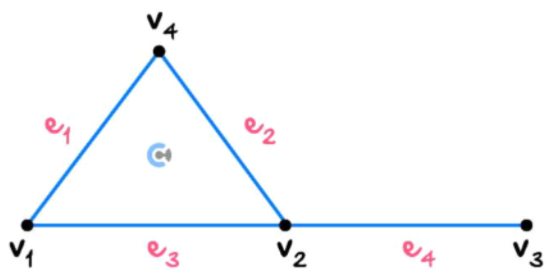


How do I cover all the roads exactly once?

Is that even possible?

- What Krish is trying to achieve is called Euler trail.
- Euler Trail is a walk of a graph where all the edges (roads) are used exactly once.

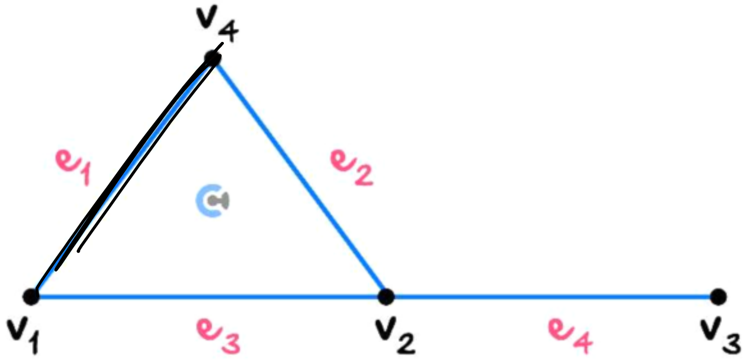
Euler Trail



- Euler Trail is a walk of a graph where all the edges are used exactly once.
- Whether a vertex is used exactly once is irrelevant.

Question 6 Walkthrough.

Consider a graph with an Euler trail.



State all possible Euler trails.

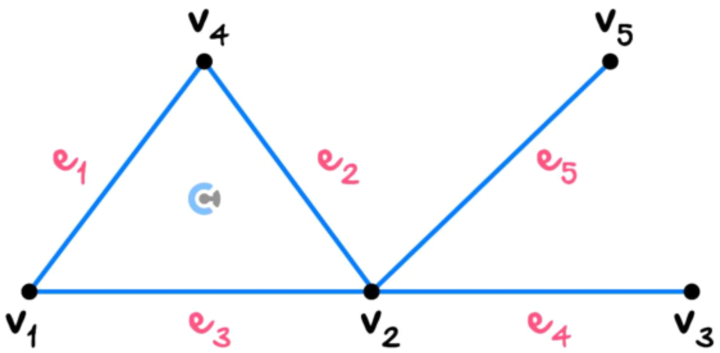
V_2, V_1, V_4, V_2, V_3 V_2, V_4, V_1, V_2, V_3

NOTE: A vertex can be visited twice. We are focusing on the edge being used exactly once.

Space for Personal Notes

Question 7

Consider a graph with an Euler trail.



State all possible Euler trails.

$V_5, V_2, V_4, V_1, V_2, V_3$

$V_3, V_2, V_1, V_4, V_2, V_5$

Space for Personal Notes

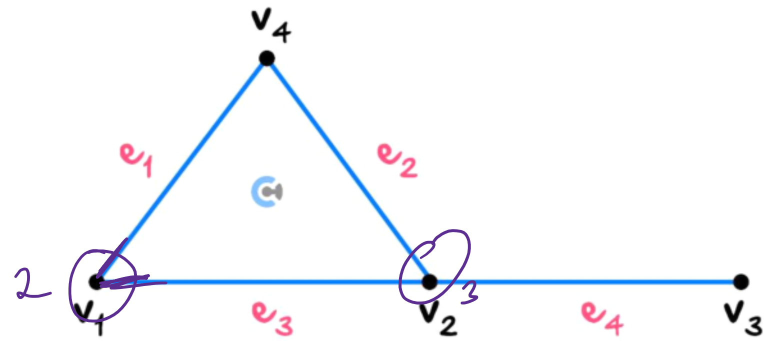
Sub-Section: Identifying Whether a Graph has an Euler Trail

Discussion: Does every graph have a valid Euler trail?

No!

Let's take a look at how we can tell if a graph has an Euler trail!

Existence of an Euler Trail



- A graph needs to satisfy one of the following two rules:

  1. Every vertex has an even degree.

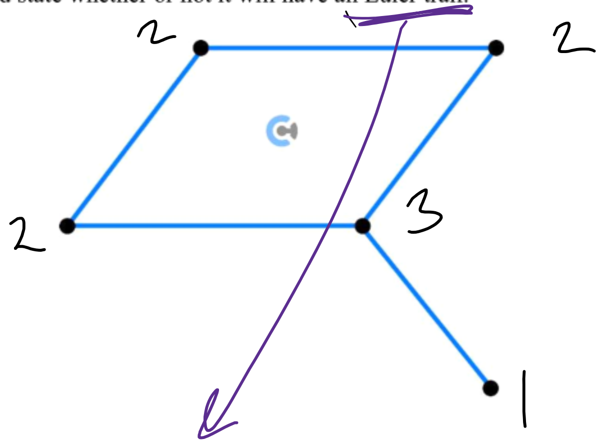
     OR

  2. Exactly two vertices have an odd degree.

Space for Personal Notes

Question 8 Walkthrough.

Consider the graph below and state whether or not it will have an Euler trail.



Yes. 2 odd degrees

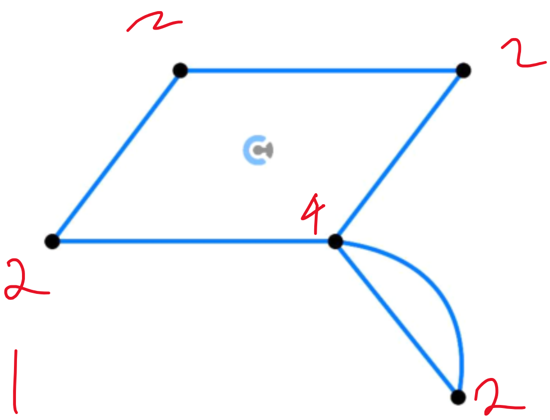
TIP: This can get very confusing. Always write down the degree of each vertex first.

Space for Personal Notes

Question 9

Consider the graphs below and state whether or not they will have an Euler trail.

a.

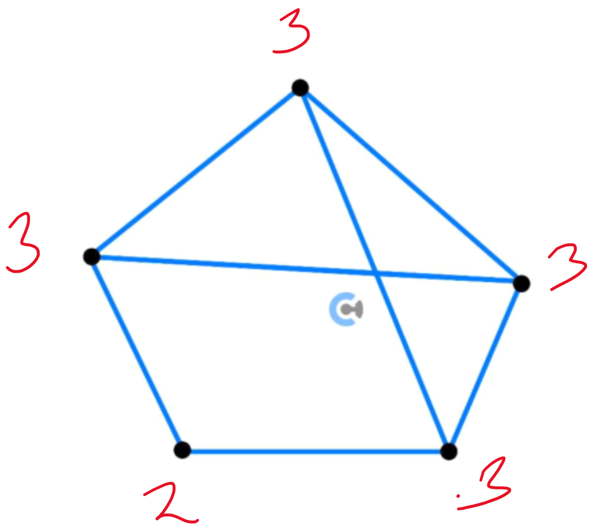


Yes!

b.

No!

>2 odd degrees.



Space for Personal Notes

Sub-Section: Finding Euler Trails Efficiently

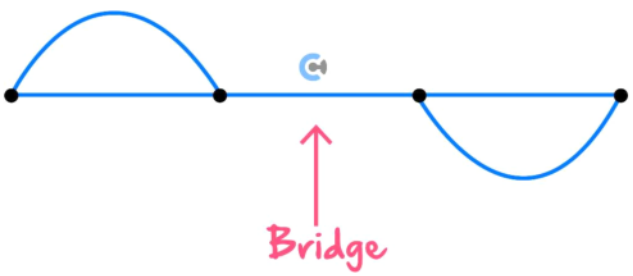
Is there any more efficient way of finding the Euler trail?

Fleury's Algorithm

- Allows us to find Euler's trail efficiently in graphs with Euler trails.
- The steps are as follows:
  1. If there are two vertices of odd degrees, then start from one of them. Otherwise, start from any vertex.
  2. Move from the current vertex across an edge to an adjacent vertex. Always choose a non-bridge edge unless there is no alternative.
  3. Delete the edge that you have just traversed.
  4. Repeat from Step 2 until there are no edges left.

Wait, what's a bridge?

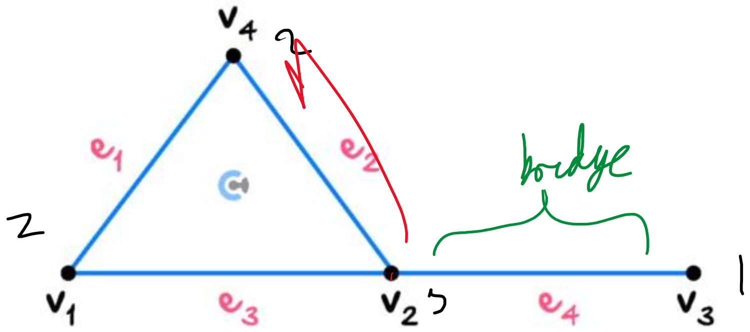
Bridge



- A bridge is an edge that, if you delete it, will cause a graph to be cut into two graphs.

Question 10 Walkthrough.

Consider a graph with an Euler trail.



State any possible Euler trail using Fleury's algorithm.

V_2, V_4, V_1, V_2, V_3

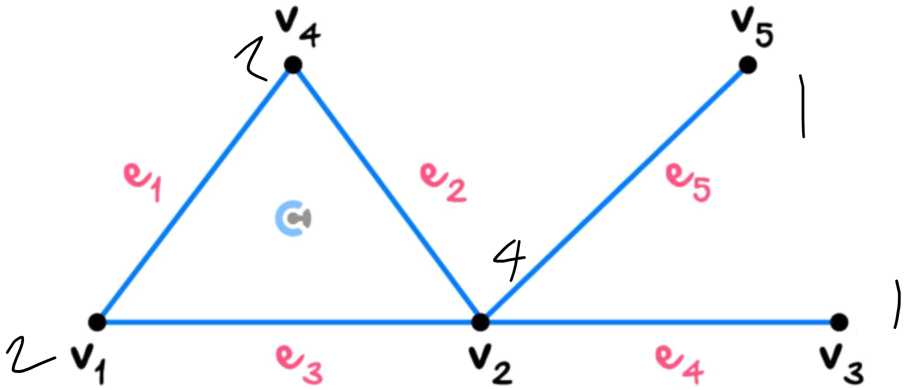
Avoiding bridge

NOTE: A vertex can be visited twice. We are focusing on the edge being used exactly once.

Space for Personal Notes

Question 11

Consider a graph with an Euler trail.



State any possible Euler trail using Fleury's algorithm.

V_5, V_2, V_4, V_1, V_2, V_3

(odd degree)

Avoiding bridge

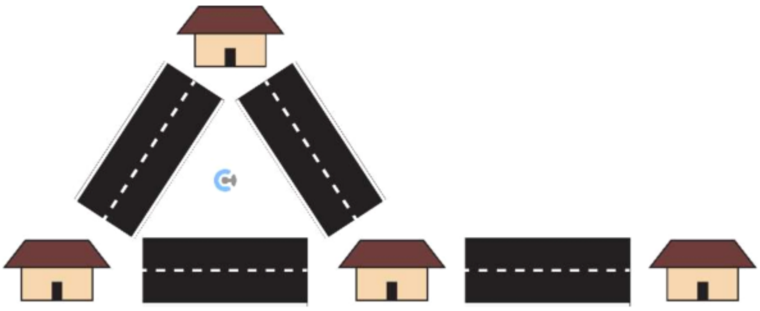
NOTE: Hope you got the same answer as before!

Space for Personal Notes

Sub-Section: Definition of Euler Circuit

Analogy: Understanding Euler Circuits

- Krish is back with his dilemma.



How do I cover all the roads exactly once?

Is that even possible?

- Krish realises his mum wants him to also come back home!

  And now asks the question,

Is there a way to cover the roads exactly once AND come back to the same position at the start?

- Krish now wants to form an Euler trail that also comes back to the same starting point.

  We call this euler circuit

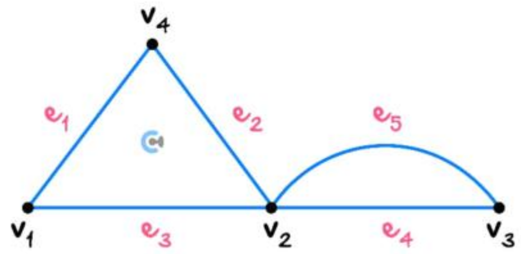
Discussion: Is the Euler circuit still an Euler trail?

Euler Circuit → Euler Trail

Euler Trail ~~→~~ Euler Circuit

Definition

Euler Circuits



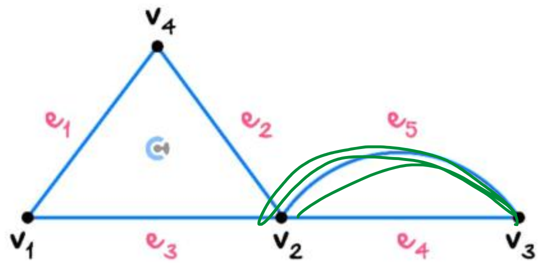
- Euler Circuit is a walk of a graph where all the edges are used exactly once.

  Starting vertex and ending vertex must be the same.

- Whether a vertex is used exactly once is irrelevant.
- Euler circuits are a subset of Euler trails.

Question 12 Walkthrough.

Consider a graph with an Euler circuit:



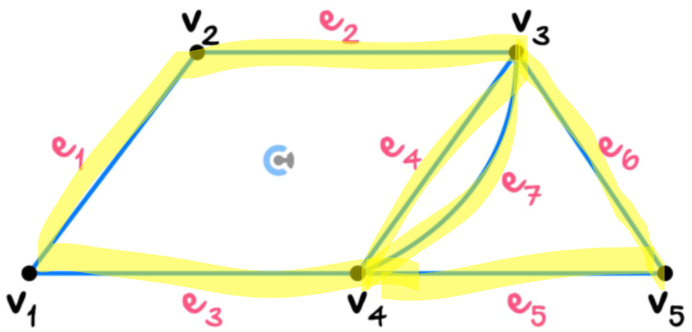
State a possible Euler circuit.

$V_3, e_5, V_2, V_4, V_1, V_2, e_4, V_3$

NOTE: We must come back to the same position.

Question 13

Consider a graph with an Euler circuit:



State a possible Euler circuit.

$V_5, V_3, e_7, V_4, e_4, V_3, V_2, V_1, V_4, V_5$

Space for Personal Notes

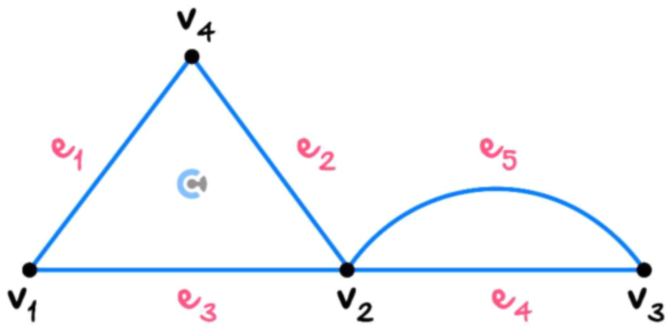
Sub-Section: Identifying Whether a Graph has an Euler Circuit

Discussion: Does every graph have a valid Euler circuit?

NO!

Let's take a look at how we can tell if a graph has an Euler circuit!

Existence of an Euler Circuit



- A graph needs to satisfy the following ~~rule~~:
  - Every vertex has an even degree.

NOTE: It's simply the first condition of the Euler trail (but not the second).

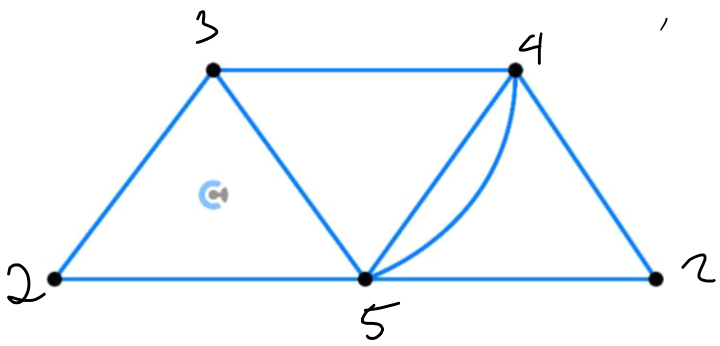
Space for Personal Notes

Question 14

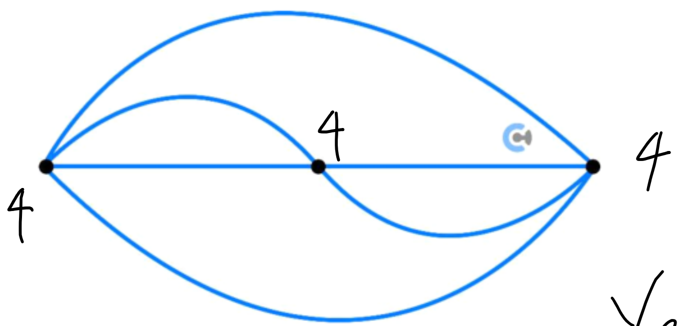
Consider the graphs below and state whether or not they will have an Euler circuit.

a.

No!



b.



Yes!

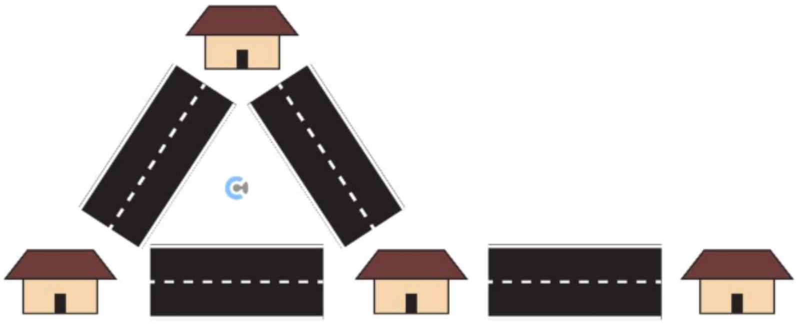
Space for Personal Notes

Section D: Hamiltonian Paths and Cycles (12C)

Sub-Section: Definition of Hamiltonian Paths

Analogy: Understanding Hamiltonian Paths

- Krish is faced with a dilemma.



How do I cover all the houses exactly once?

Is that even possible?

- What Krish is trying to achieve is called Hamiltonian path.
- Hamiltonian path is a walk of a graph where all the vertices(houses) are used exactly once.

NOTE: Similar to Euler, but we are now looking at vertices instead of edges.

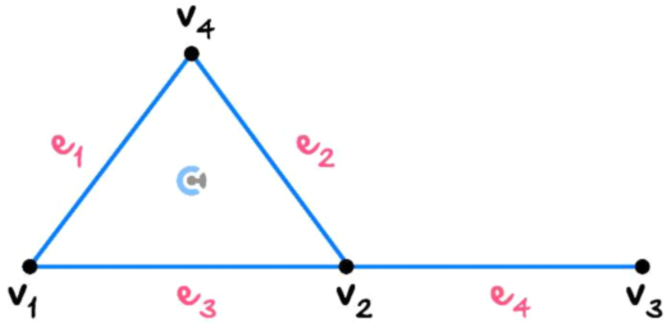
Hamiltonian Path

- A path that covers all the vertices exactly once.

Space for Personal Notes

Question 15 Walkthrough.

Consider a graph with a Hamiltonian path:



State any possible Hamiltonian path.

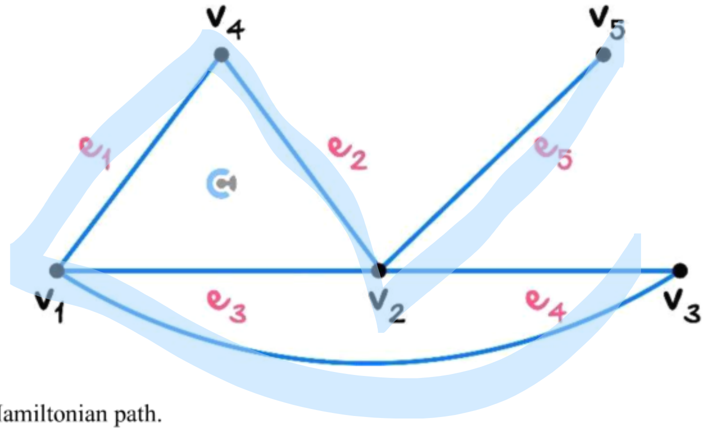
V_3, V_2, V_4, V_1

NOTE: An edge can be visited twice. We are focusing on the vertex being visited exactly once.

Space for Personal Notes

Question 16

Consider a graph with a Hamiltonian path:



State any possible Hamiltonian path.

V5, V2, V4, V1, V3

Space for Personal Notes

Sub-Section: Definition of Hamiltonian Cycles

Discussion: What do you think the Hamiltonian cycle means?

Vertices once, start and finish are same

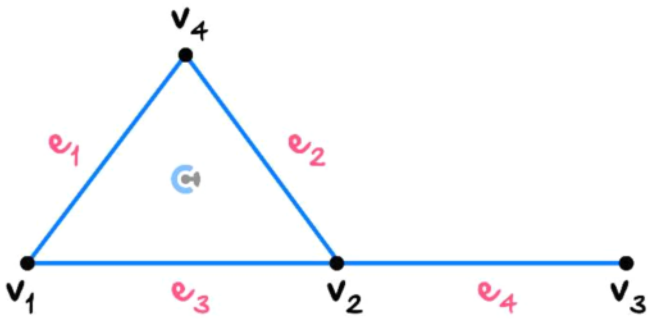
Hamiltonian Cycle

- A path which covers all the vertices exactly once (Except the starting one).

  The starting position and ending vertex must be identical.

Question 17 Walkthrough.

Consider the graph below:



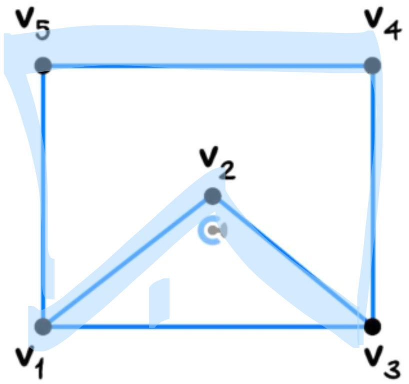
Does it have a Hamiltonian cycle? If so, state below.

No.

Space for Personal Notes

Question 18

Consider the graph below:



Does it have a Hamiltonian cycle? If so, state below.

Yes

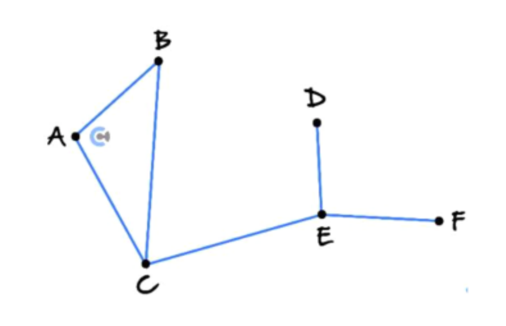
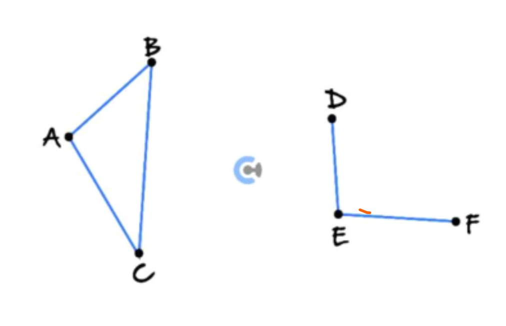
$V_1, V_2, V_3, V_4, V_5, V_1$

Space for Personal Notes

Section E: Trees

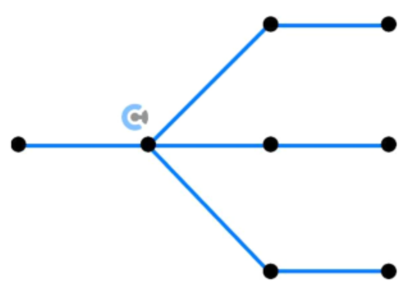
Sub-Section: Definition of Trees

REMINDER: Connected Graphs

Connected Graph	Disconnected Graph
	

- A connected graph is a graph where it is possible to reach all vertices by moving along edges.

Trees

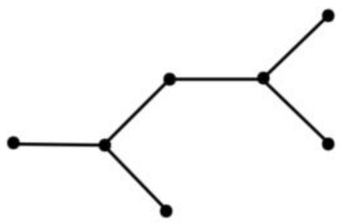


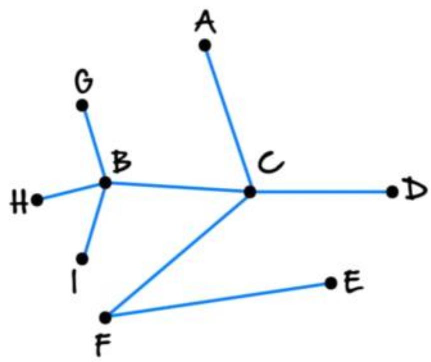
- A tree is a connected graph (can go from one vertex to any other vertex) without any cycle.
- We cannot come back to the same vertex by not repeating the edges.

Space for Personal Notes

Definition

Trees



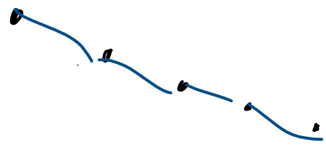


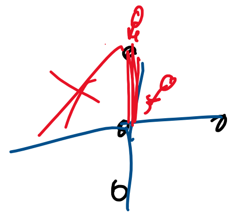
- A tree is a connected graph (can go from one vertex to any other vertex) without any cycle.
- We cannot come back to the same vertex by not repeating the edges.
- **Equivalent conditions for a tree:**
  - A simple graph with n vertices is a tree if ANY of the following conditions are met:
    - It is connected and contains no cycles.
    - It is connected and has $n-1$ edges.
    - It has no cycles and has $(n-1)$ edges.
    - It is connected but would become disconnected if any edge is removed.
    - It is connected and would form a cycle if any edge is added.
    - Any two vertices are connected by only a single path.

Space for Personal Notes

Question 19 Walkthrough.

Sketch any tree that contains five vertices.

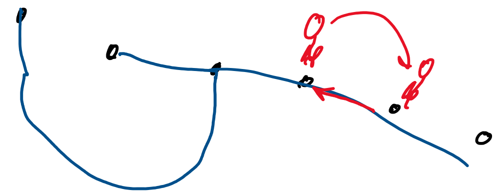




Space for Personal Notes

Question 20

Sketch any tree that contains six vertices.

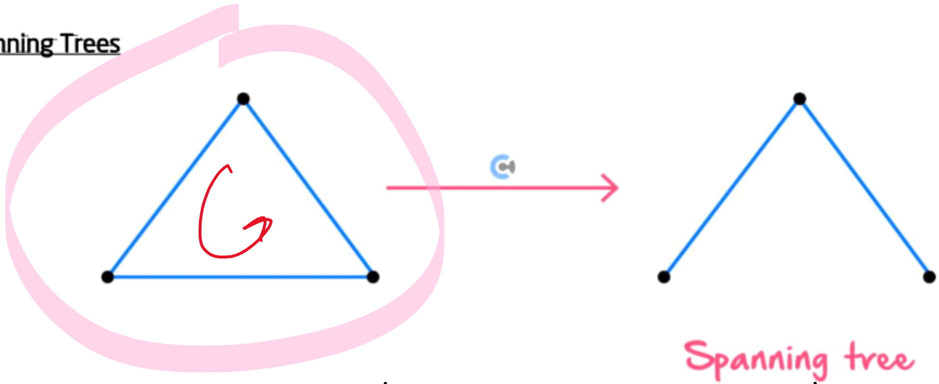


Space for Personal Notes

Sub-Section: Spanning Tree

What is a spanning tree?

Spanning Trees



- Spanning trees are sub-graphs that are trees and retain all of the vertices of the original graph.

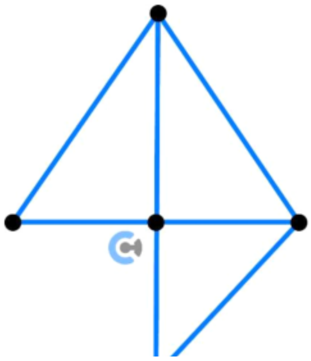
  The original graph does not have to be a tree.

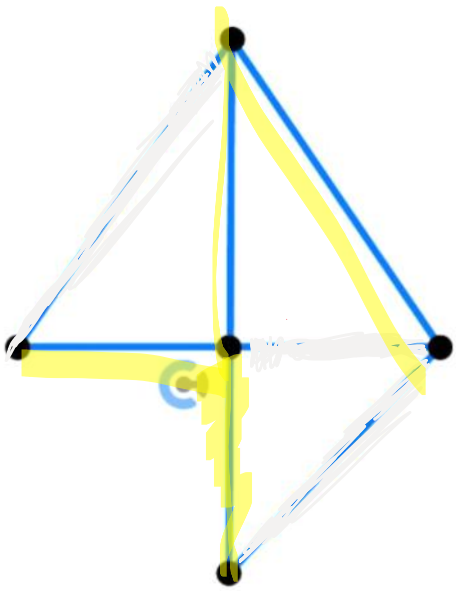
NOTE: The cycle goes through all the points and comes back to the same point.

Space for Personal Notes

Question 21 Walkthrough.

Find the spanning tree for the given graph below:



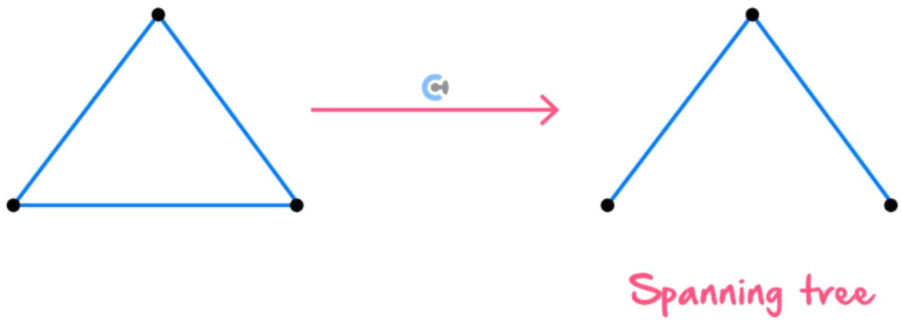


Space for Personal Notes

Is there a systematic way to find a spanning tree for harder questions?

Definition

Algorithm for Finding a Spanning Tree

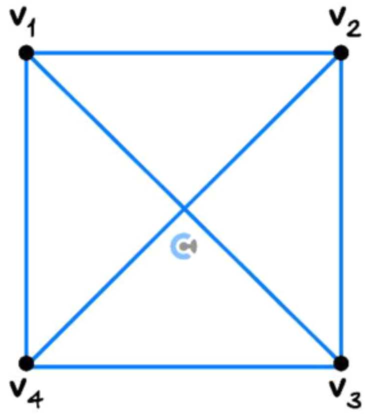


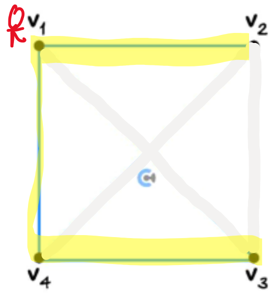
- The steps in finding a spanning tree:
  1. If the graph has no cycles, then stop.
  2. Choose **any** edge that belongs to a cycle, and delete the chosen edge.
  3. Repeat from 1.

Space for Personal Notes

Question 22 Walkthrough.

Using the algorithm, find a spanning tree for the given graph below:

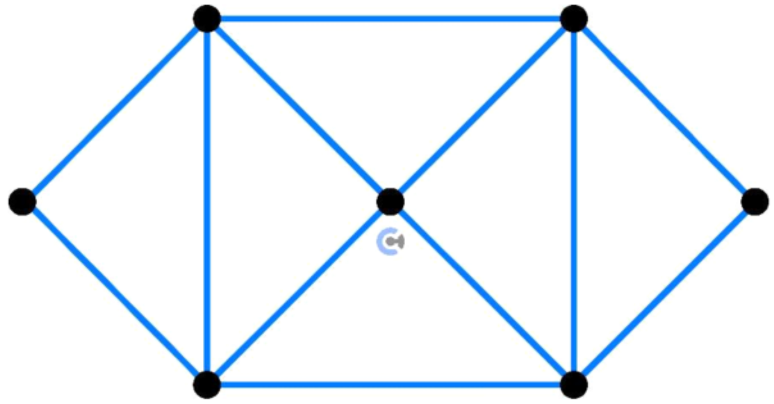


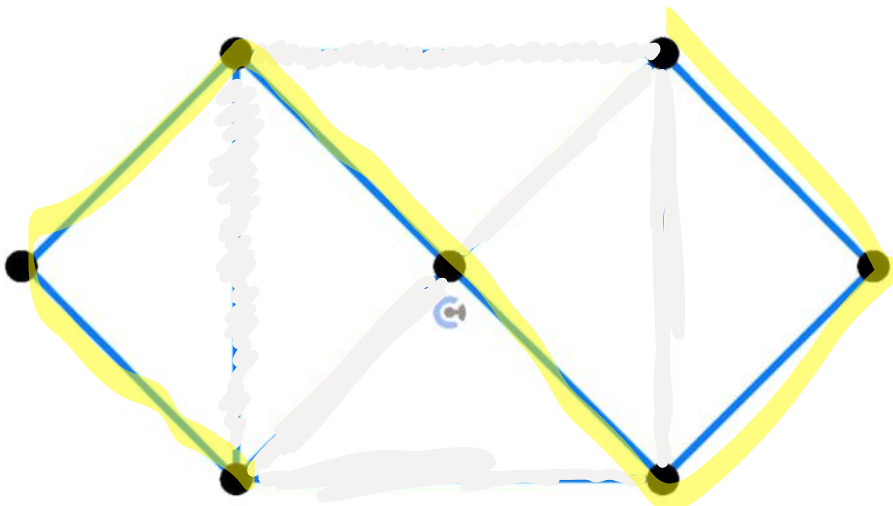


Space for Personal Notes

Question 23 (4 marks) **Walkthrough.**

Using the algorithm, find a spanning tree for the given graph below:





Space for Personal Notes

Cheat Sheet

[5.4.1] – Walk of a graph

- **Walk of a Graph**

$$v_1, e_1, \cdots, v_{n-1}, e_1, v_n$$

- Order of the vertices (points) and edges (lines) we are visiting.
- Alternates between a vertex and an edge.
- If there is only one edge between two vertices, stating that an edge can be abbreviated.

- **Length of a Walk**

$$\text{Length of a Walk} = \text{Number of Edges in a Walk}$$

- Number of edges in a walk.
- If we use the same edge multiple times, we count it multiple times for the length.

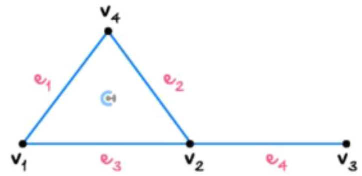
- **Number of Walks with the Length of n**

$$A^n = \begin{bmatrix} v_1, v_1 & v_1, v_2 & v_1, v_3 \\ v_1, v_2 & v_2, v_2 & v_2, v_3 \\ v_1, v_3 & v_3, v_2 & v_3, v_3 \end{bmatrix}$$

- For a walk of length n, we power the adjacency matrix by n.
- The element gives the number of walks of length n between two vertices.

[5.4.2] – Euler trail and circuits

- **Euler Trail**



- Euler Trail is a walk of a graph where all the edges are used exactly once.
- Whether a vertex is used exactly once is irrelevant.

- **Existence of an Euler Trail**

- A graph needs to satisfy one of the following two rules:

1. Every vertex has an even degree.

OR

2. Exactly two vertices have an odd degree.

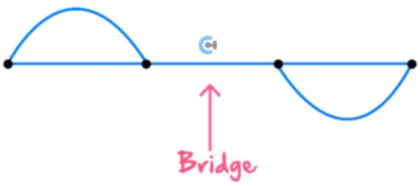
- **Fleury's Algorithm**

- Allows us to find Euler's trail efficiently in graphs with Euler trails.
- The steps are as follows:

1. If there are two vertices of odd degrees, start from one. Otherwise, start from any vertex.
2. Move from the current vertex across an edge to an adjacent vertex. Always choose a non-bridge edge unless there is no alternative.
3. Delete the edge that you have just traversed.
4. Repeat from Step 2 until there are no edges left.

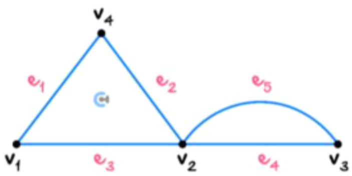
Cheat Sheet

- **Bridge**



A bridge is an edge that, if you delete it, will cause a graph to be cut into two graphs.

- **Euler Circuits**



Euler Circuit is a walk of a graph where all the edges are used exactly once.

The starting vertex and ending vertex must be the same.

Whether a vertex is used exactly once is irrelevant.

Euler circuits are a subset of Euler trails.

- **Existence of an Euler Circuit**

A graph needs to satisfy the following rule:

- Every vertex has an even degree.

[5.4.3] – Hamiltonian Paths and Cycles

- **Hamiltonian Path**

A path that covers all the vertices exactly once.

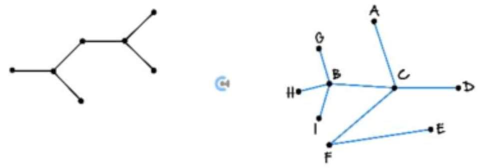
- **Hamiltonian Cycle**

A path that covers all the vertices exactly once. (Except the starting one.)

The starting position and ending vertex must be identical.

[5.4.4] – Trees

- **Trees**



A tree is a connected graph (can go from one vertex to any other vertex) without any cycle.

We cannot come back to the same vertex by not repeating the edges.

- **Equivalent Conditions for a Tree**

A simple graph with n vertices is a tree if ANY of the following conditions are met:

- It is connected and contains no cycles.
- It is connected and has $(n-1)$ edges.
- It has no cycles and has $(n-1)$ edges.
- It is connected but would become disconnected if any edge is removed.
- It is connected and would form a cycle if any edge is added.
- Any two vertices are connected by only a single path.

CONTOUREDUCATION

Website: contoureducation.com.au | **Phone:** 1800 888 300 | **Email:** hello@contoureducation.com.au

VCE Specialist Mathematics ½

Free 1-on-1 Consults

What Are 1-on-1 Consults?

- **Who Runs Them?** Experienced Contour tutors (45 + raw scores and 99 + ATARs).
- **Who Can Join?** Fully enrolled Contour students.
- **When Are They?** 30-minute 1-on-1 help sessions, after-school weekdays, and all-day weekends.
- **What To Do?** Join on time, ask questions, re-learn concepts, or extend yourself!
- **Price?** Completely free!
- **One Active Booking Per Subject:** Must attend your current consultation before scheduling the next. :)

SAVE THE LINK, AND MAKE THE MOST OF THIS (FREE) SERVICE!

Booking Link

bit.ly/contour-specialist-consult-2025