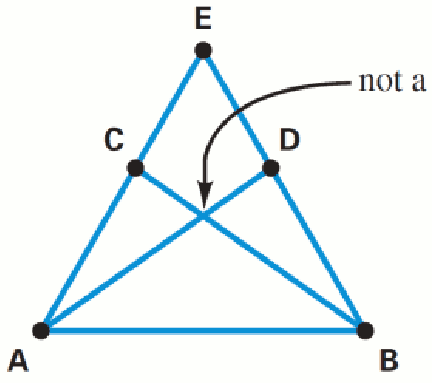
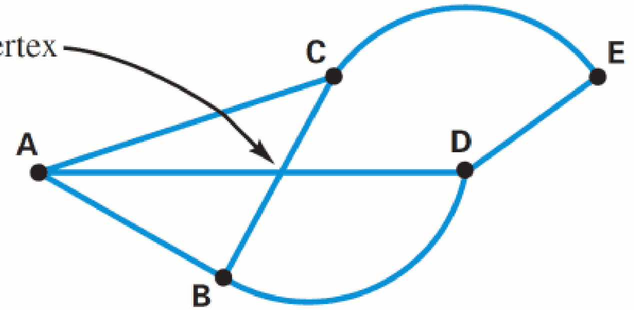
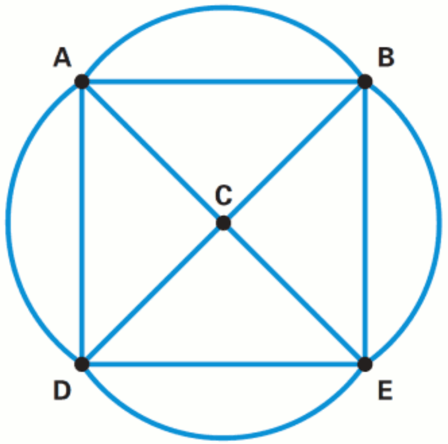
MAT110 REVIEW

Graph Theory

DEFINITIONS



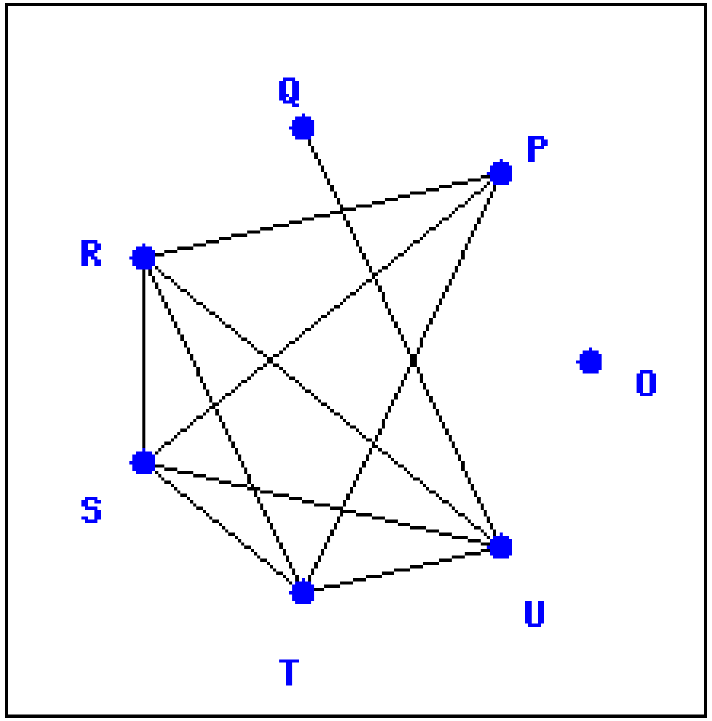




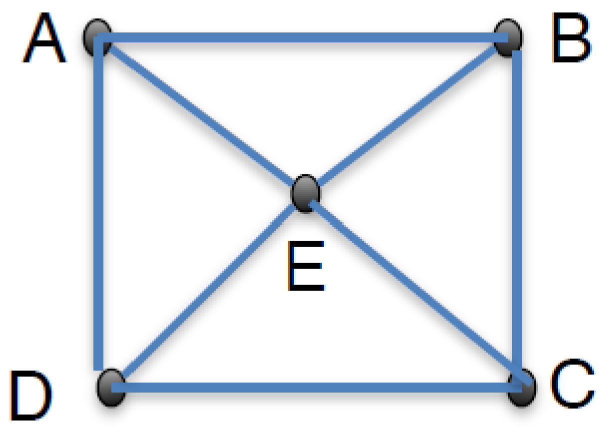
- A **graph** consists of a set of a points, called **vertices**, and lines, called **edges**, that join pairs of vertices.
- The **degree** of a vertex is the number of edges that use that vertex as an endpoint.
- A **simple graph** has no loops and no pair of edges are joined by more than one edge.
- A graph is **connected** if there exists some path between every pair of vertices.
- A **bridge** in a connected graph is an edge such that if it were removed the graph would no longer be connected.
- A **component** is a maximally connected subgraph of the original graph.

EXAMPLE

1. What is the degree of vertex R?
   - Degree 4
2. What is the degree of vertex O?
   - Degree 0
3. How many components does the graph have?
   - 2 components
4. Ignoring vertex O, identify a bridge in the graph.
   - Edge QU is a bridge
5. Is this a simple graph?
   - Yes



DEFINITIONS



Paths

- A **path** is a sequence of distinct edges of the graph such that the ending vertex of one edge is the beginning vertex of the next edge in the sequence.
- One possible path in the graph above:
  ADEB

Circuits

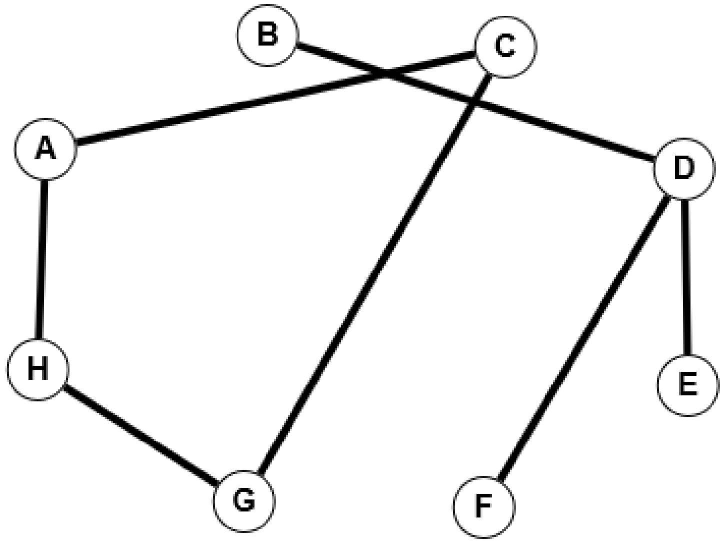
- A **circuit** is a path that begins and ends at the same vertex.
- One possible circuit in the graph above:
  AECDA

- *Note: The number of edges in a path (or circuit) is called its **length**.*

EXAMPLE

- Sketch a simple graph with 8 vertices that contains a circuit of length 4, a vertex of degree 3, and 2 components. Identify the vertex of degree 3 and the circuit of length 4 in your graph.

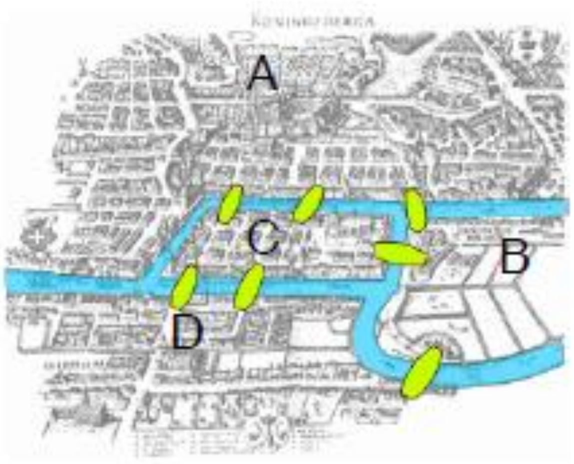
- Circuit of length 4: AHGCA
- Vertex of degree 3: D



EULER

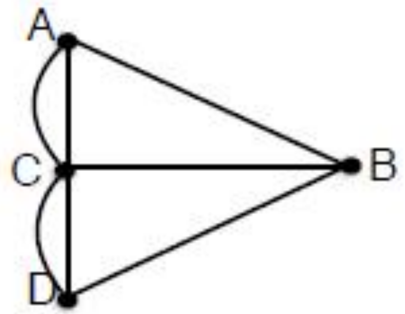
Euler Path

- An Euler path is a path that uses every edge of the graph exactly once.



Euler Circuit

- An Euler circuit is an Euler path that begins and ends at the same vertex.



EULER'S THEOREM

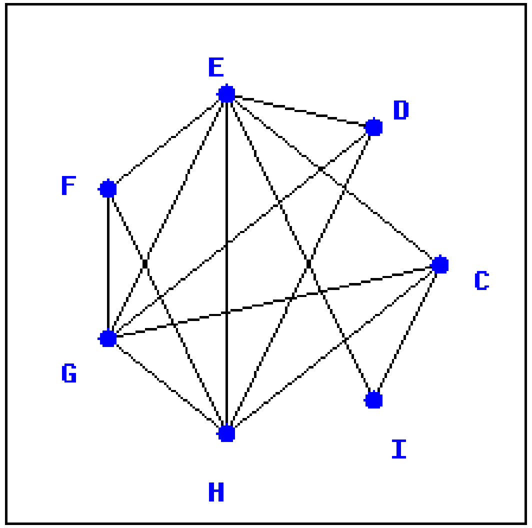
- If a graph has **more that two** vertices of odd degree, then it has no Euler paths.
- If a graph is connected and has **exactly two** vertices of odd degree, then it has at least one Euler path. The path must start at one of the odd-degree vertices and end at the other.
- If a connected graph has **zero** vertices of odd degree, then it has at least one Euler circuit

# Odd Vertices	Euler Path?	Euler Circuit?
0	***YES**	***YES**
2	***YES**	No
4, 6, 8 . . .	No	No
1, 3, 5 . . .	*No Such Graphs Exist!*	

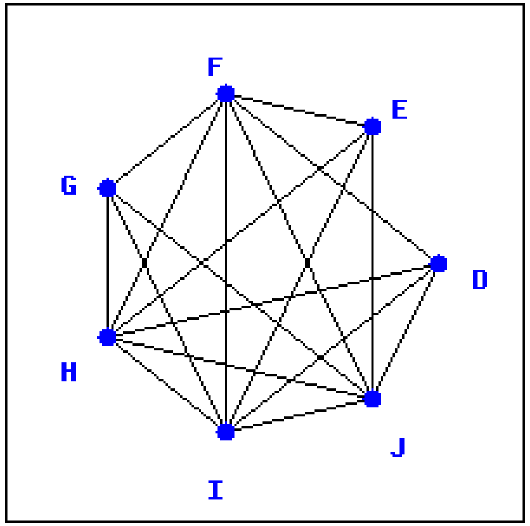
**As long as the graph is connected.*

EXAMPLE

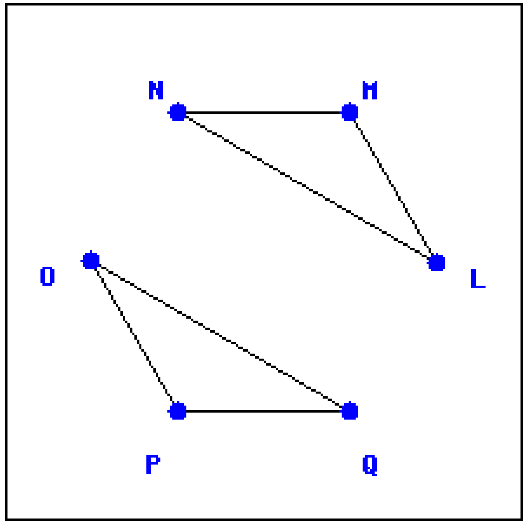
- Determine whether the following graphs have an Euler path, Euler circuit, or neither.



Neither!
There are 4 odd vertices; D, F, G, H



Euler circuit!
This is a connected graph with no odd vertices.



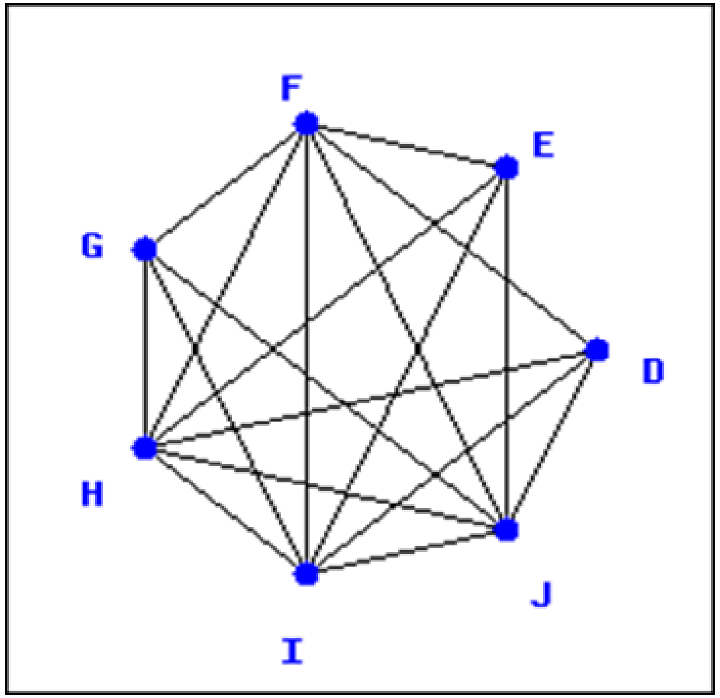
Neither!
This graph is not connected.

FLEURY'S ALGORITHM

- Given a connected graph with all even vertices, choose any vertex to start. Travel along edges in the graph according to the following rules:
  1. After traveling through an edge, erase it (or highlight it).
  2. You may only travel over a bridge if there is no other option.

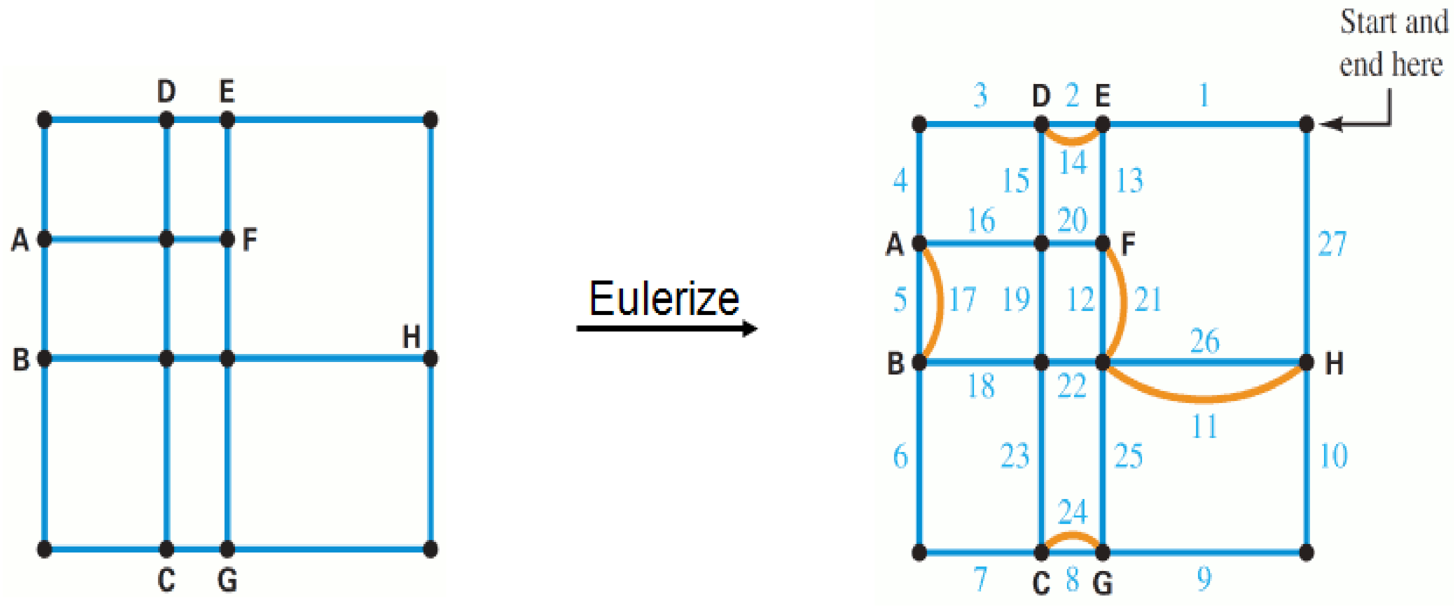
One possible Euler circuit for the graph to the right:

D F E J D I J H G I E H F G J F I H D



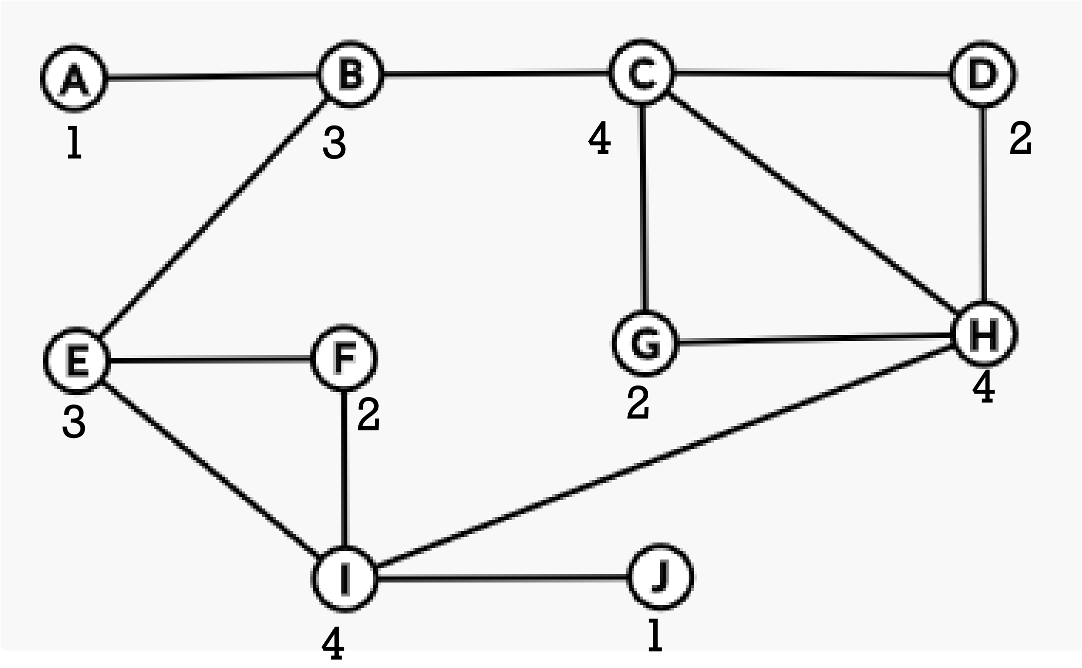
EULERIZING A GRAPH

- Eulerizing a graph allows us to find an Euler circuit by making copies of existing edges so that all vertices have even degree.
  - Remember, you cannot create new edges, only repeat existing edges.
  - Start by connecting vertices of odd degree, keep repeating edges until all vertices have even degree.



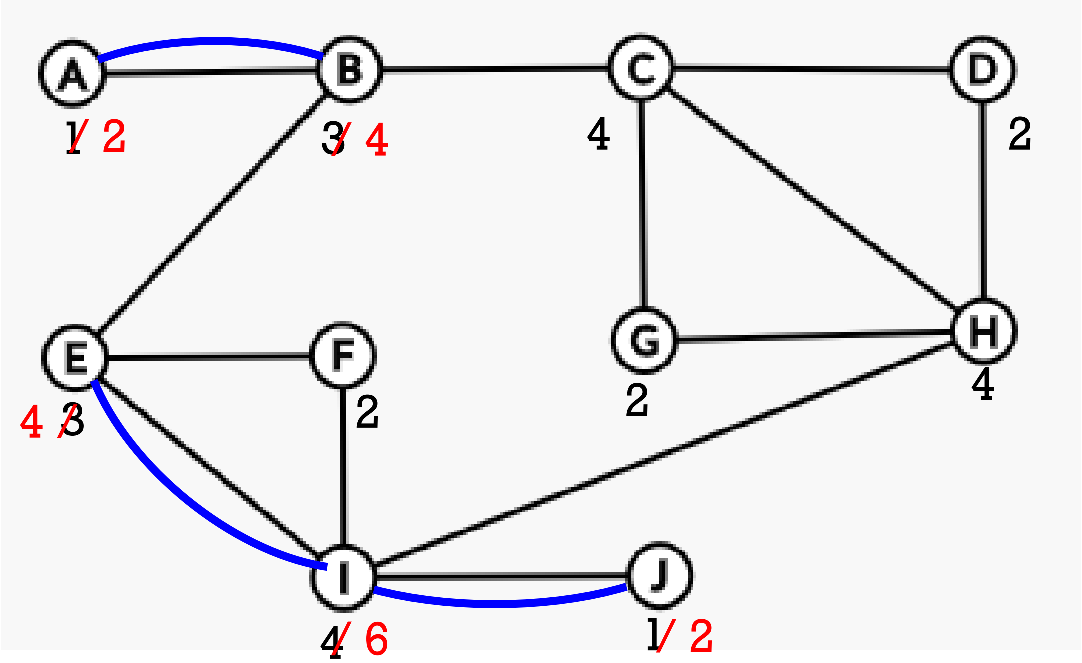
EXAMPLE

- Find a minimal Eulerization of the graph below:



EXAMPLE

- Find a minimal Eulerization of the graph below:



- One possible circuit is: A B C D H G C H I J I F E I E B A

HAMILTON

Hamilton Path

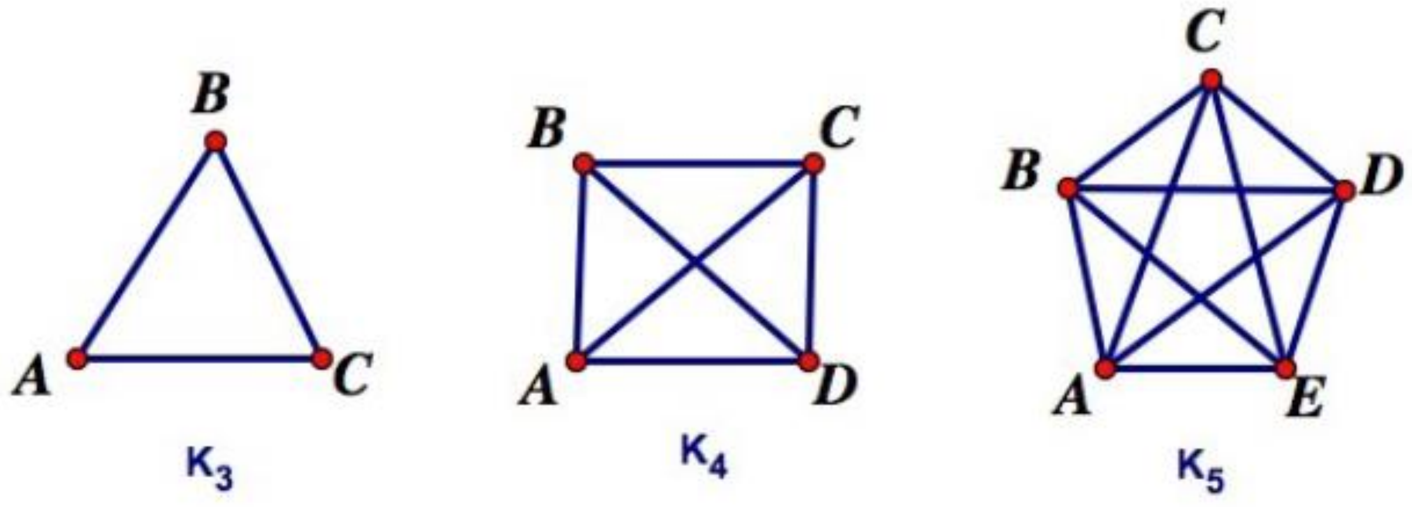
- A Hamilton path passes through all vertices exactly one time.

Hamilton Circuit

- A Hamilton circuit is a Hamilton path that begins and ends at the same vertex.

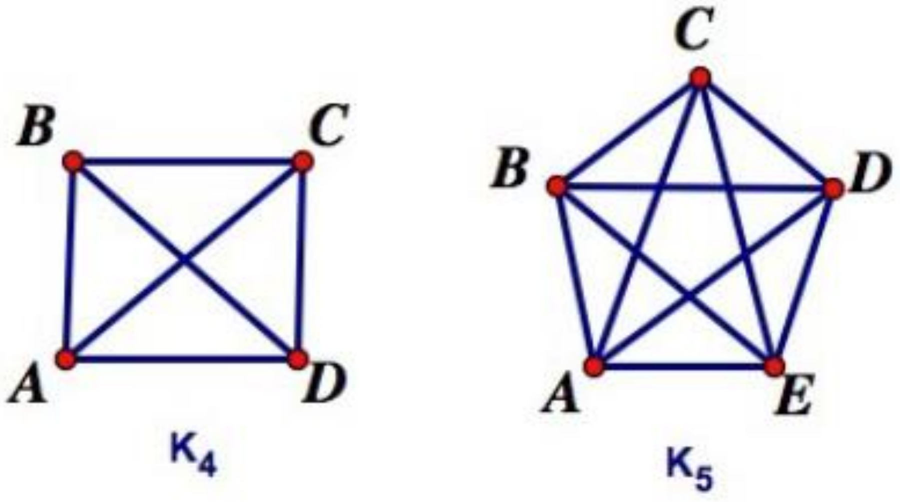
DEFINITIONS

- A complete graph is one in which every pair of vertices is joined by an edge. A complete graph with n vertices is denoted by K_n.
- In a weighted graph, numbers, or weights, are assigned to the edges.



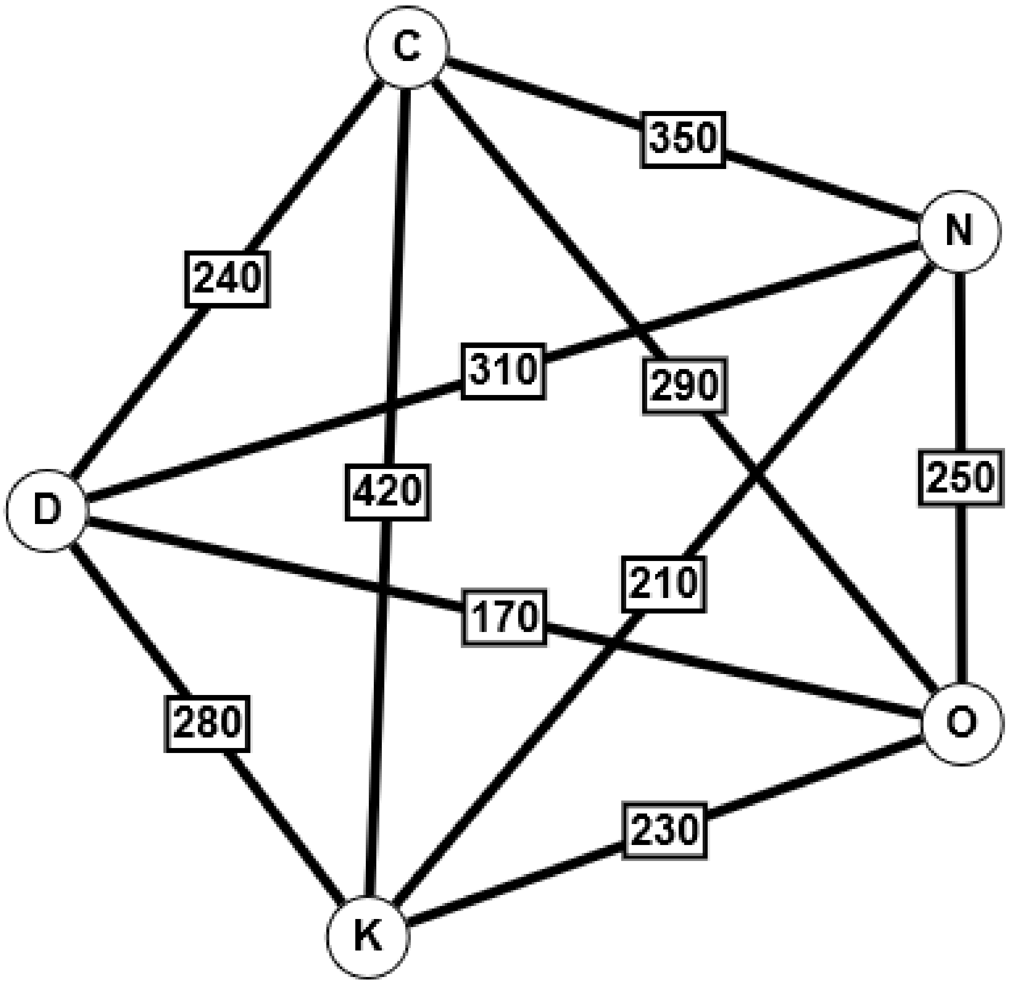
EXAMPLE

- How many Hamilton circuits are there in a complete graph with 4 vertices?
  - $(n - 1)! = 3! = \mathbf{6}$
- How many Hamilton Circuits are there in a complete graph with 5 vertices, if you don't count reversals?
  - $\frac{(n-1)!}{2} = \frac{4!}{2} = \frac{24}{2} = \mathbf{12}$



TRAVELING SALESPERSON PROBLEM

- The vertices of the graph represent cities, and the weighted edges of the graph represent the travel cost between cities.
- Griff needs to visit Chicago, New York, Orlando, and Denver before coming home to Kansas City. Use the weighted graph to find the sequence of cities for Griff to visit that will minimize his total travel cost.



BRUTE FORCE ALGORITHM

- List all the Hamilton circuits in the graph
- Find the weight of each circuit
- The circuits with the smallest weights give the solution.

- KODCNK and KNCDOK have the smallest weights, so these two trips would minimize Griff's travel costs.

Note: Generally we will only count the number of circuits that would need to be checked if we were to use Brute Force Algorithm. For a K_n, there are $(n-1)!$ total Hamilton circuits.

Hamilton Circuit	Weight
KOCDNK	1280
KOCNDK	1460
KODCNK	1200
KODNCK	1480
KONCDK	1350
KONDCK	1450
KCODNK	1400
KCONDK	1550
KCDONK	1290
KCNODK	1470
KDOCNK	1300
KDCONK	1270

NEAREST NEIGHBOR ALGORITHM

- Step 1: Start at any vertex X.
- Step 2: Choose the smallest weighted edge connected to X. The vertex at the end of this edge is the next vertex in the circuit.
- Step 3: Continuing choosing edges as you did in step 2 without revisiting any vertices.
- Step 4: After all vertices have been chosen, close the circuit by returning to the starting vertex.

NEAREST NEIGHBOR ALGORITHM

- Starting at Kansas City, K.

KN – 210

NO – 250

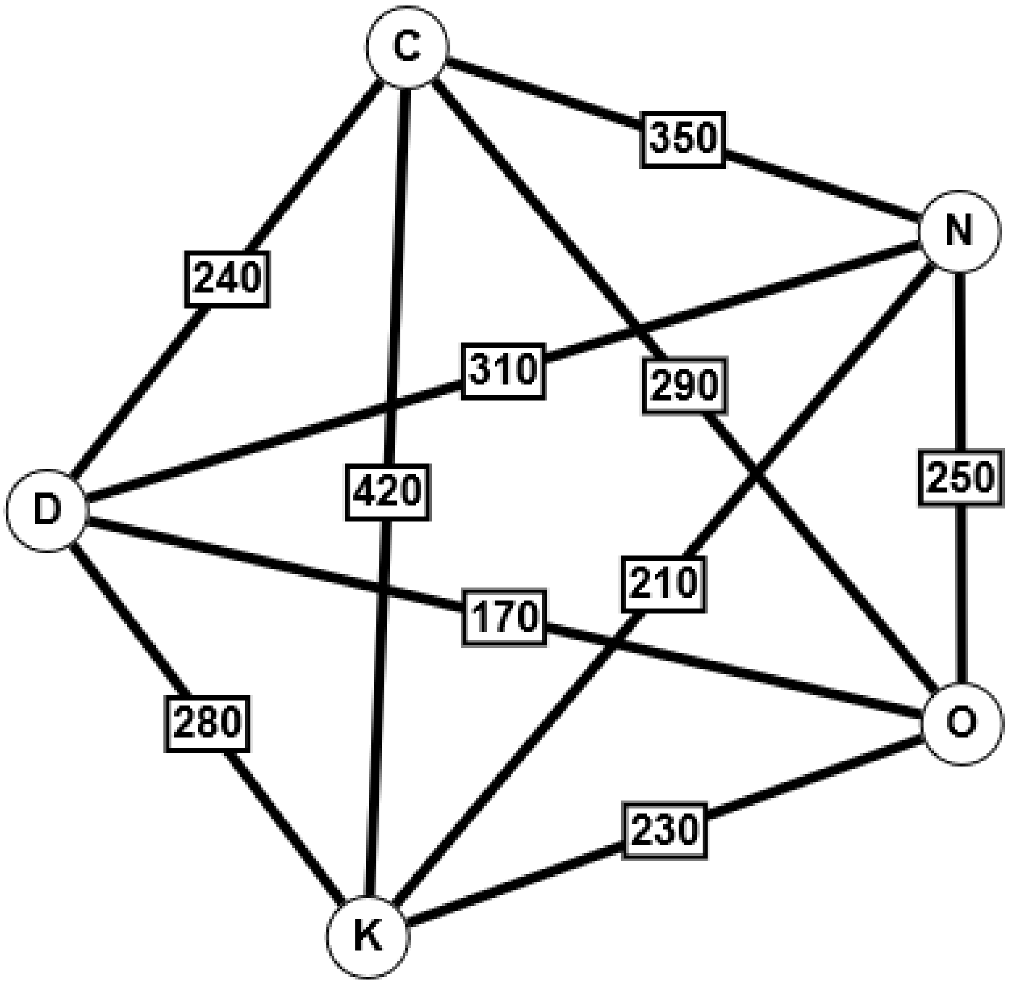
OD – 170

DC – 240

CK – 420

Total Weight: 1290

Hamilton Circuit: K N O D C K



SIDE SORTED/BEST EDGE ALGORITHM

- Step 1: Choose the edge with the smallest weight.
- Step 2: Choose any remaining edge in the graph with the smallest weight.
- Step 3: Continue adding the edge with the smallest weight while following these conditions:
  - Do not form a circuit before all vertices have been added.
  - Do not add an edge that would give a vertex degree 3.

Note: You are not following a path as you add edges to create the circuit. You can rewrite your circuit to start at a specific vertex. Remember ABCDEA is the same circuit as CDEABC.

SIDE SORTED/BEST EDGE ALGORITHM

- Write the circuit starting at Kansas City, K.

DO – 170

NK – 210

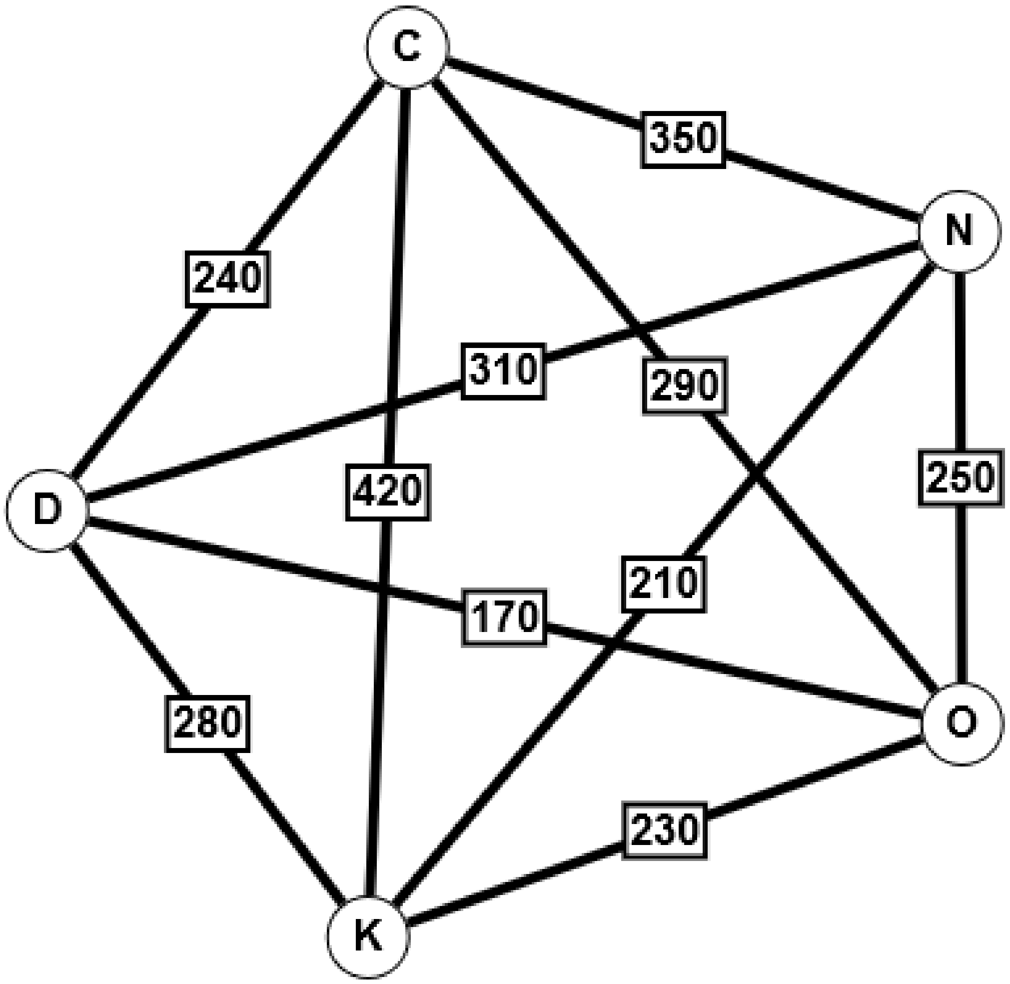
KO – 230

CD – 240

NC – 350

Total Weight: 1200

Hamilton Circuit: K N C D O K or K O D C N K



EXAMPLE

The table to the right represents the distances between each area of Disney's Magic Kingdom; Main Street(M), Adventureland(A), Frontierland(R), Liberty Square(L), Fantasyland(F), and Tomorrowland(T). We want to find the shortest route between all six areas so we can spend more time in each place.

	M	A	R	L	F	T
M		6	7	4	6	5
A	6		2	1	6	5
R	7	2		3	6	7
L	4	1	3		4	3
F	6	6	6	4		2
T	5	5	7	3	2	

EXAMPLE

- If we were to use Brute Force Algorithm to find out how many different routes are possible, how many circuits would we have to check?

	M	A	R	L	F	T
M		6	7	4	6	5
A	6		2	1	6	5
R	7	2		3	6	7
L	4	1	3		4	3
F	6	6	6	4		2
T	5	5	7	3	2	

$(6-1)! = 120$

Note: If it said without reversals, we would divide by 2.

EXAMPLE

- Use the Nearest Neighbor Algorithm, starting at Main Street, to find the most efficient route through the park that visits each area exactly once and ends back at Main Street to watch the Happily Ever After fireworks show.

ML – 4
LA – 1
AR – 2
RF – 6
FT – 2
TM – 5

Total Weight: 20
Hamilton Circuit: M L A R F T M

	M	A	R	L	F	T
M		6	7	4	6	5
A	6		2	1	6	5
R	7	2		3	6	7
L	4	1	3		4	3
F	6	6	6	4		2
T	5	5	7	3	2	

EXAMPLE

- Use the Side Sorted/Best Edge Algorithm to find the most efficient route through the park visiting every area exactly once. Make sure you write your circuit ending at Main Street for the Happily Ever After Fireworks show.

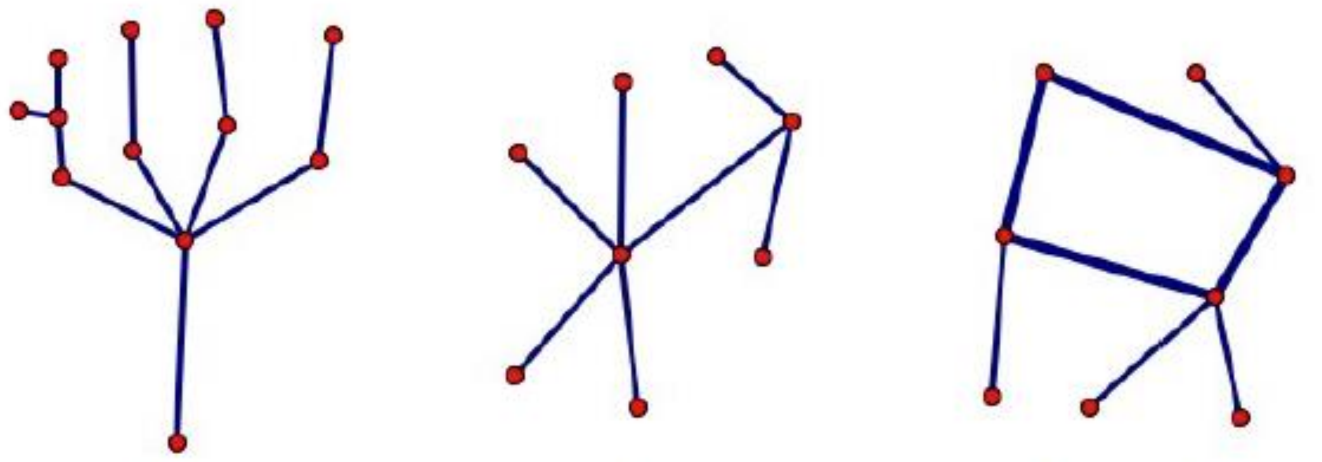
LA – 1
TF – 2
RA – 2
TL – 3
FM – 6
RM - 7

Total Weight: 21
Hamilton Circuit: MRALTFM or MFTLARM

	M	A	R	L	F	T
M		6	7	4	6	5
A	6		2	1	6	5
R	7	2		3	6	7
L	4	1	3		4	3
F	6	6	6	4		2
T	5	5	7	3	2	

TREES

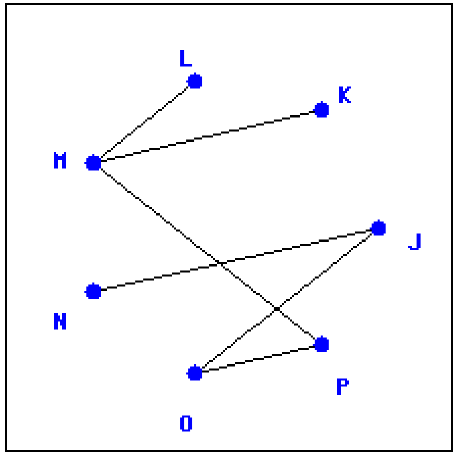
- A **tree** is a connected graph with no circuits.
- A **forest** is a collection of one or more trees.
- A **spanning tree** is a subgraph that is connected, contains all the original vertices in the graph, and has no circuits.



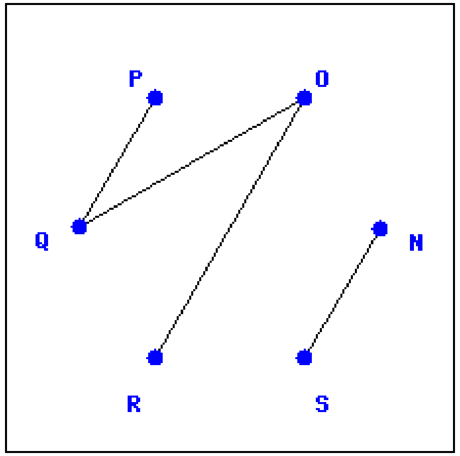
Tree Tree Not a Tree

EXAMPLE

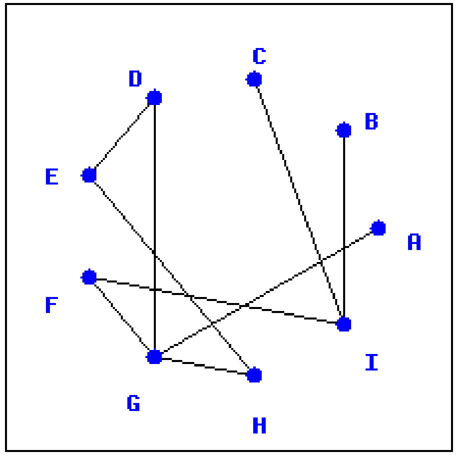
- Determine whether the following graphs are trees, forests, or neither.



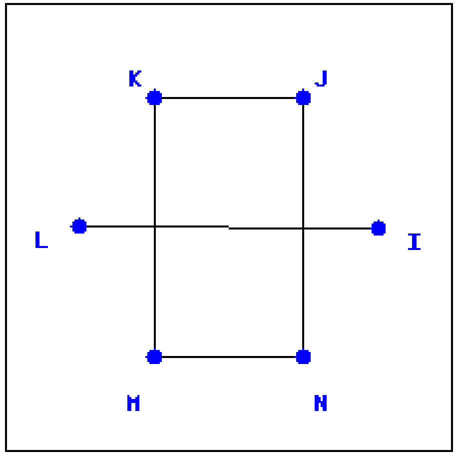
Tree!
(Also a forest)



Forest!



None!
EHGDE is a circuit.



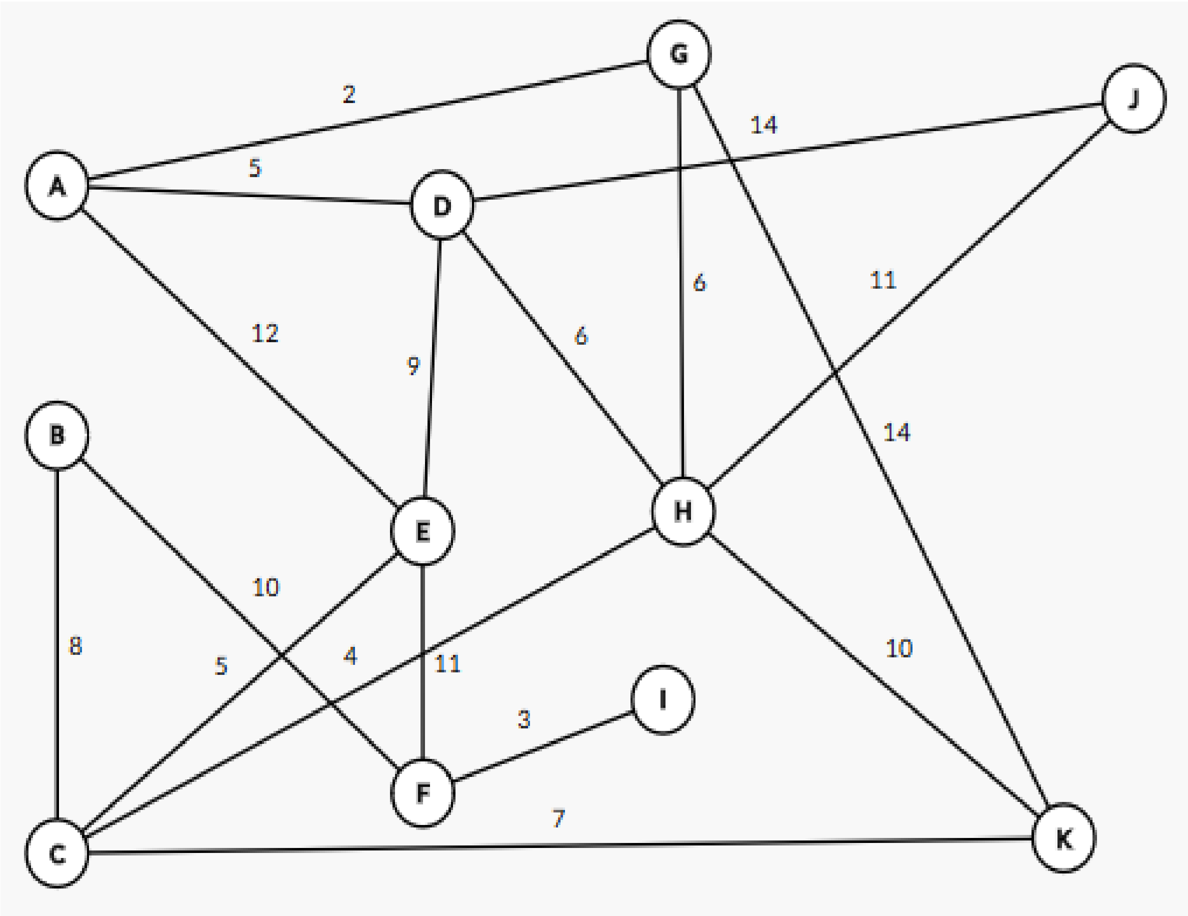
None!
KJNMK is a circuit.

KRUSKAL'S ALGORITHM

Used to find a minimal spanning tree of a connected weighted graph

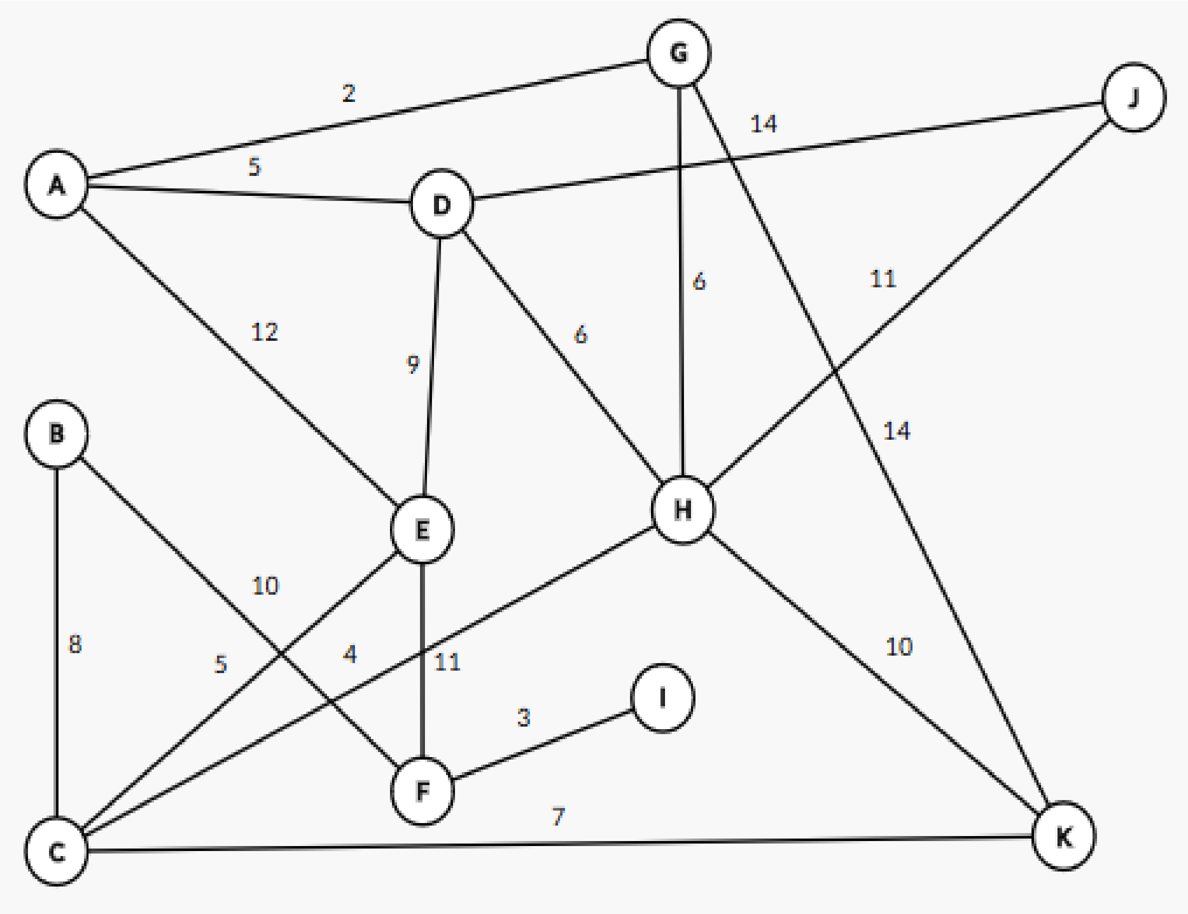
- Step 1: Choose any edge with the smallest weight.
- Step 2: Choose any remaining edge in the graph with the smallest weight.
- Step 3: Continue adding the edge with the smallest weighted edge without creating a circuit until all vertices have been added to your subgraph.

EXAMPLE



- Use Kruskal's algorithm to sketch a minimal spanning tree for the weighted graph above. Be sure to list the edges chosen in order and give the total weight of your spanning tree.

EXAMPLE



AG – 2
FI – 3
CH – 4
CE – 5
AD - 5
DH – 6
CK – 7
BC – 8
BF – 10
HJ – 11

All original vertices are connected, so these edges form a minimal spanning tree.

Note: Kruskal's algorithm does not find circuits. All that is required is that all vertices are included.

CENTER FOR ACADEMIC SUPPORT

Hearnes 213

816-271-4524