Graph Theory

1. Define the terms below.

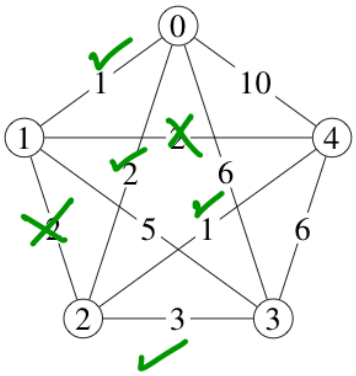
   (a) An Euler circuit is a circuit that includes every edge exactly 1 time.

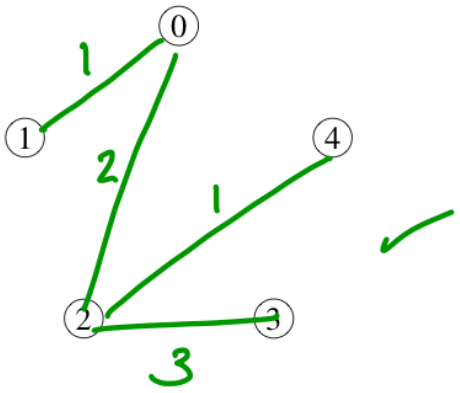
   (b) An Euler path is a path that includes every edge exactly 1 time.

   (c) A Hamiltonian circuit is a circuit that includes every vertex exactly 1 time.

   (d) A spanning tree is a tree (no circuits) that includes every vertex.

2. (a) Use Kruskal's Algorithm to find a minimum weight spanning tree in the graph below.





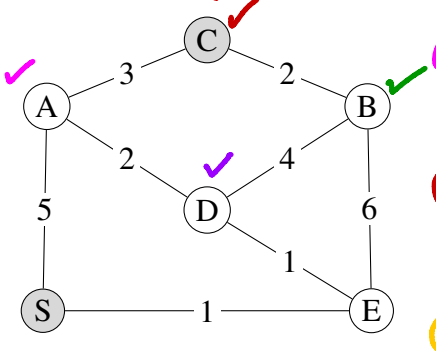
   (b) Give an example of a real-world problem which you would want to find a minimum weight spanning tree. (You would need to state what are the vertices, edges, and weights!)

vertices: houses
edges: water lines
weights: cost of installation

The minimum weight spanning tree is the cheapest way to connect all houses to water lines.

3. (a) Use Dijkstra's Algorithm to find a shortest path from vertex *S* to vertex *C*.

start D. Alg. here



vertex	current/ visited	**tentative** minimum distance to C	preceding vertex
A	~~C~~ V	3	C
B	~~C~~ V	2	C
C	~~C~~ V		
D	~~C~~ V	~~6~~ 5	~~B~~ A
E	~~C~~ V	~~8~~ 6	~~B~~ D
S	C	~~8~~ 7	E

things to check!

3 A, D
2 D, E
1 A, D
4 E
5 S
6

done

Shortest S-C path: S E D A C

length: 7

(b) Give an example of a real-world problem which you would want to find the minimum distance from *S*. (You would need to state what the vertices, edges, and weights represent and what is the significance of the shortest path in this context.)

Vertices: cities
edges: roads connecting cities
weights: distance

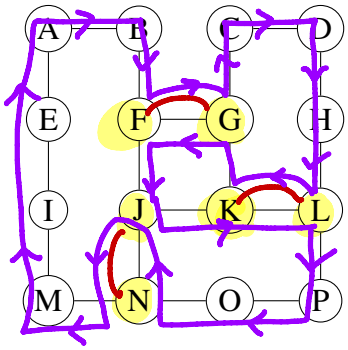
Dijkstra's Algorithm gives the shortest distance from your town (S) to all other towns

4. (a) What does it mean to Eulerize a graph? What is your goal?

You duplicate edges in order to ensure the graph contains an Euler circuit.

You need to make all vertices of odd degree into ones with even degree. All the vertices with even degree need to still be even.

(b) Eulerize the graph below by adding the fewest number of edges and then find an Euler circuit in the resulting graph.



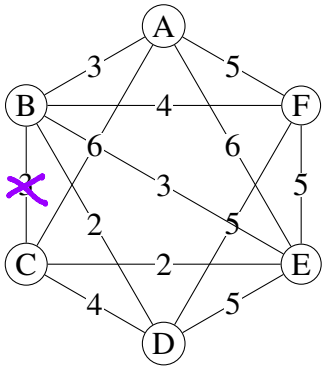
○ Vertices of odd degree
• Added edges
• Euler circuit.

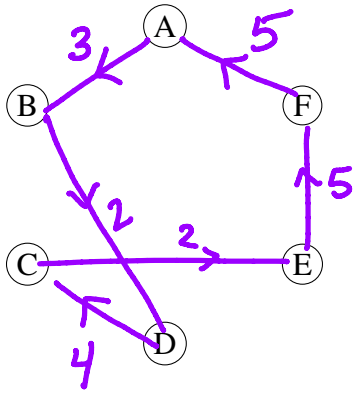
A B F G C D H L K G F J K L P O J N M I E A

(c) Give an example of a real-world problem which you would want to find an Euler circuit. (You would need to state what the vertices, edges, and weights represent and what is the significance of the shortest path in this context.)

Vertices: road intersections
edges: roads
An Euler circuit would be a way to efficiently plow the streets, so you only go over a street as few times as possible.

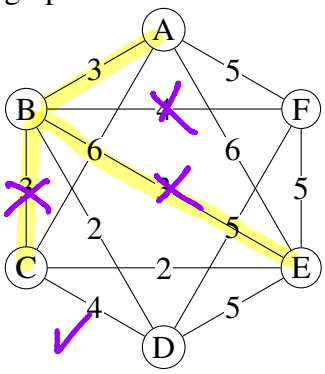
5. (a) Use the Nearest Neighbor Algorithm starting a vertex A, to find a Hamiltonian Circuit in the graph below and determine its weight.

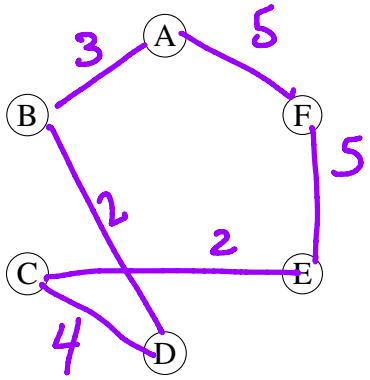




weight 3+2+4+2+5+5 = 21

(b) Use the Cheapest Link Algorithm to find a minimum weight Hamiltonian Circuit in the graph below.





Weight = 21

(c) Give an example of a real-world problem which you would want to find a minimum weight Hamiltonian circuit. (You would need to state what the vertices, edges, and weights represent and what is the significance of the shortest path in this context.)

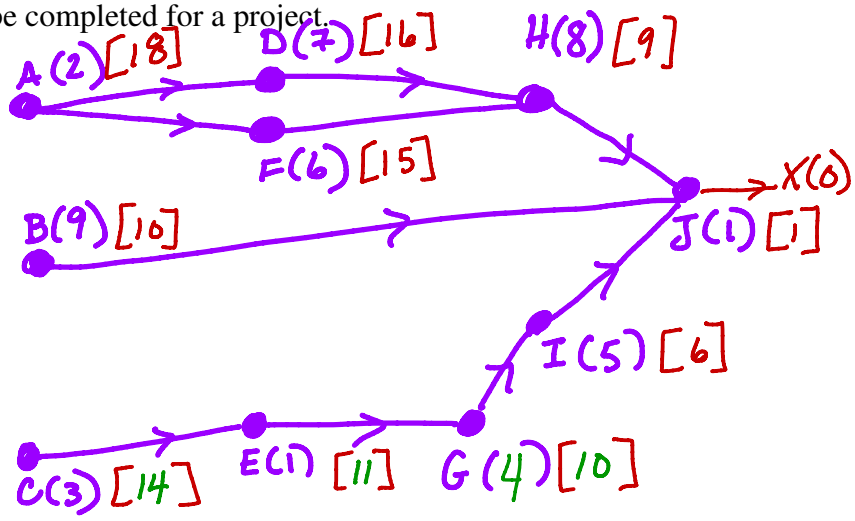
Vertices: cities
edges: roads
weight: distance

A minimum weight Hamiltonian circuit would be the cheapest route to visit every vertex exactly 1 time.

Scheduling

6. The table below contains the tasks to be completed for a project.

task	time (in minutes)	dependencies/ prerequisites
A	2	
B	9	
C	3	
D	7	A
E	1	C
F	6	A
G	4	E
H	8	D,F
I	5	G
J	1	B,H,I



(a) To the right, above, create a digraph representing the project.

(b) Find a critical path and its corresponding critical time. Explain the significance of the critical time.

weight(CEGIJ) = 3+1+5+5+1 = 15

weight(ADHJ) = 2+7+8+1 = 18 ✓

ADHJ critpath with weight 18

Significance: This project takes a MINIMUM of 18 minutes.

(c) Create a priority list using the decreasing time algorithm. (That is, create a priority list ordered by decreasing time.)

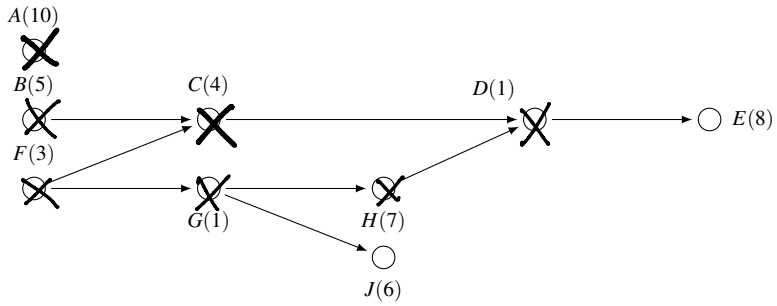
B, H, D, F, I, G, C, A, E, J

(d) Use the Backflow algorithm to assign critical numbers to each vertex in the graph above.

(e) Create a priority list using the critical time algorithm. (That is, create a priority list ordered by decreasing **critical** time.)

A, D, F, C, E, G, B, G, H, I, J

7. Use the diagraph below and the given priority list to construct a schedule using two processors.



Construct a schedule using the priority list

time	ready	done
0	A, B, F	
3	A, G	F
4	A, J, H	G
5	G, J, H	B
9	J, H	C
14	J	A
16	D	H
17	E	D
20		J
25		E