Spring 2025 **Math F113X**

Exam 2

Name: Solutions **Section:** ☐ **10:30 am (Leah Berman)**
☐ **2:15pm (Jill Faudree)**

Rules:

- Partial credit will be awarded, but you must show your work.
- You may have 1/2 of a standard page of paper ($8.5'' \times 5.5''$) of notes, both sides.
- Calculators are allowed.
- Place a box around your FINAL ANSWER to each question where appropriate.
- Turn off anything that might go beep during the exam.

Good luck!

Problem	Possible	Score
1	9	
2	9	
3	12	
4	14	
5	14	
6	10	
7	12	
8	12	
9	8	
Extra Credit	(5)	
Total	100	

1. (9 points)

Define the following:

a. A Hamiltonian circuit is a circuit that includes every vertex exactly one time

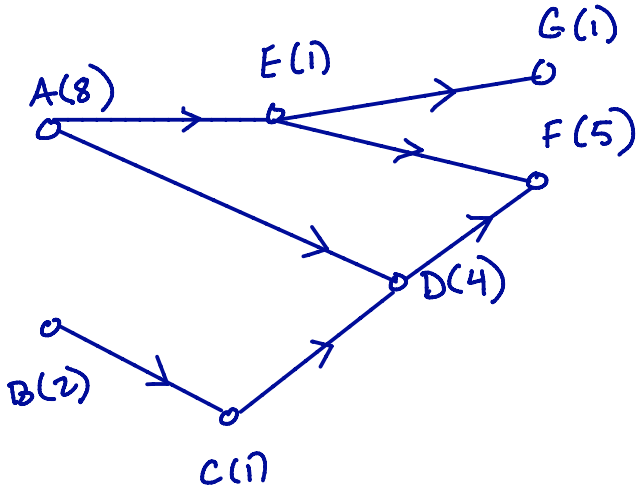
b. An Euler circuit is a circuit that includes every edge exactly one time

c. a spanning tree is a connect graph that includes every vertex but has no circuits.

2. (9 points)

Construct a scheduling digraph corresponding to the following list of tasks and dependencies.

Task	Time	dependency
A	8 hours	
B	2 hours	
C	1 hour	B
D	4 hours	A, C
E	1 hour	A
F	5 hours	D, E
G	1 hour	E



3. (12 points)

Recall Kruskal's Algorithm says the following:

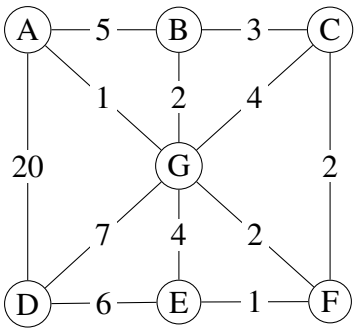
Kruskal's Algorithm: Select the cheapest edge in the graph that does not create a circuit. Stop when a spanning tree is obtained.

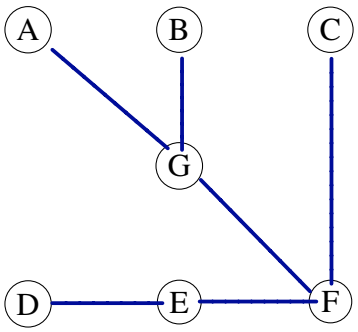
a. Use Kruskal's Algorithm to determine a **minimum cost spanning tree** in the following graph.

Break any ties by choosing the edge that comes earlier in the alphabet. (For example, edge DC is the same edge as edge CD, and CD alphabetizes earlier than GH.)

Construct a minimum cost spanning tree in the second graph, and keep track of the steps of the algorithm, the edges that you are using and the weights, in the table below.

For convenience, the edges of the graph are listed in order in the table below.





Sorted edges	weight	used?
AG	1	✓
EF	1	✓
BG	2	✓
CF	2	✓
FG	2	✓
BC	3	✗
CG	4	✗
EG	4	✗
AB	5	✗
DE	6	✓
DG	7	
AD	20	

done

$1+1+2+2+2+6$

$= 14$

b. What is the total cost of the spanning tree you found? 14

4. (14 points)

Recall:

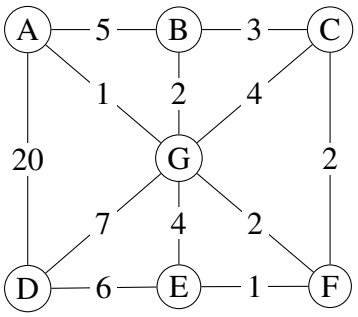
Sorted Edges / Cheapest Link Algorithm: Select the cheapest edge in the graph that does not create a vertex of degree 3 or close the circuit too soon.

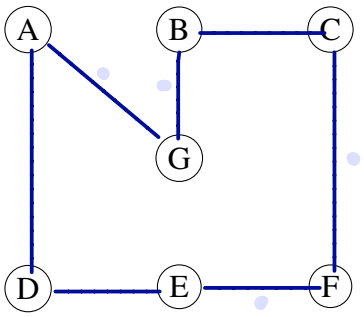
a. Use **Sorted Edges / Cheapest Link Algorithm** to find a **Hamiltonian circuit** in the following graph.

Break any ties by choosing the edge that comes earlier in the alphabet. (For example, edge DC is the same edge as edge CD, and CD alphabetizes earlier than GH.)

Keep track of the cycle in the second graph.

For convenience, the edges of the graph are listed in order in the table below.





Sorted edges	weight	used?
AG	1	✓
EF	1	✓
BG	2	✓
CF	2	✓
FG	2	✗
BC	3	✓
CG	4	✗
EG	4	✗
AB	5	✗
DE	6	✓
DG	7	✗
AD	20	✓

1+1+2+2+6+20+3

b. Write the Hamiltonian circuit you found, beginning with vertex A.

A G BC F E D A

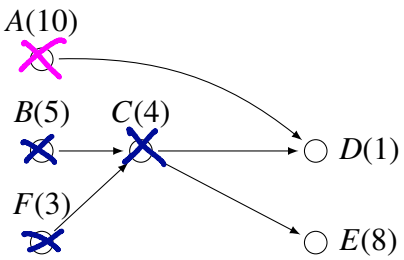
c. What is the total weight of the Hamiltonian circuit? 35

d. Is this the cheapest possible Hamiltonian circuit in this graph? No Explain your answer below.

Any circuit that avoids AD would be cheaper.
For example: ABCFEDGA has weight 25.

7. (14 points)

Consider the following digraph. The units for the tasks are hours.

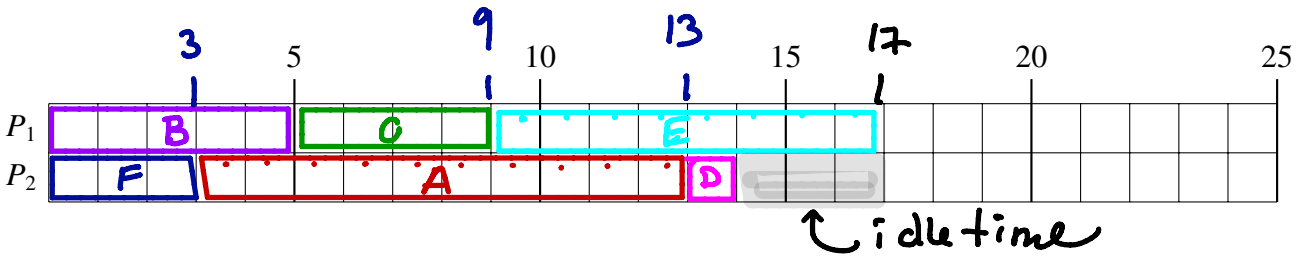


time	ready	done
0	A, B, F	
3	A	F
5	C	B
9	E	C
13	D	A

a. Construct a schedule for two processors corresponding to the following digraph, given the priority list

B, C, D, A, E

Make sure to label the tasks in the schedule. You may use the table to the right of the digraph to track your work.



b. How much idle time is in this schedule, and when? 3 hours at the end

c. How long did this schedule take to compute? 17 hours

d. What is the critical path and the critical time for this digraph?

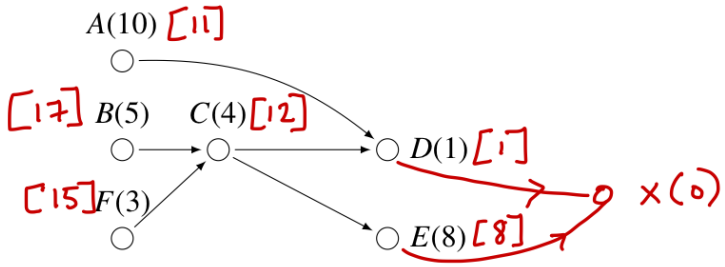
Critical path B C E Critical time 17

e. Is the schedule you found above optimal? Explain your answer: how do you know?

Yes. The finishing time is the critical time

6. (10 points)

Consider the following digraph:



a. Construct the priority list for the digraph corresponding to the **decreasing time algorithm.**

A, E, B, C, F, D

b. Label the vertices of the digraph according to the **backflow algorithm.**

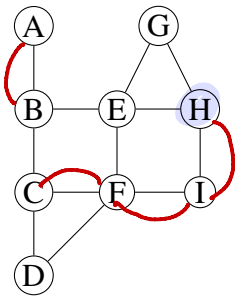
c. Construct the priority list for the digraph corresponding to the **critical path algorithm**.

B, F, C, A, E, D

7. (12 points)

a. Explain why the graph on the right does not contain an Euler circuit.

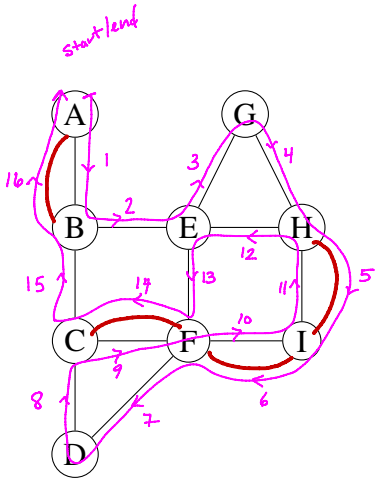
Vertex H has odd degree.



b. Eulerize the graph on the right **using as few edge duplications as possible.** Make your added edges very clear!

(I made paths between vertices of odd degree.)

c. Find an **Euler circuit** in the eulerized graph by **drawing** the circuit on the (larger) graph to the right and **listing** the vertices of the circuit in the blank below. (You will have to add your edges from part b in again!)



Euler circuit: ABE GHIFDCFIHEFCBA

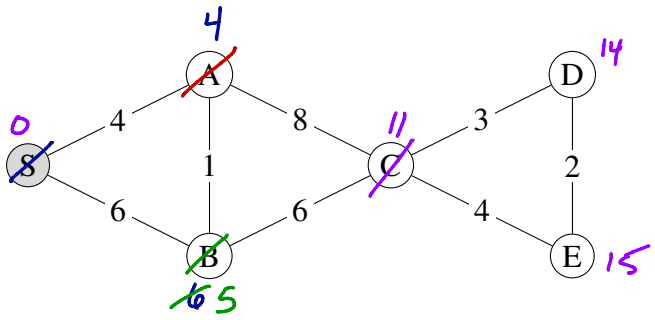
8. (10 points)

Recall Dijkstra's algorithm says the following:

Dijkstra's Algorithm

input: a graph with distances (weights) on the edges and a starting vertex, say s
output: the shortest distance between s and every vertex in the graph
rough strategy: All vertices get **tentative** distances to vertex s. One-by-one, vertices are explored and tentative distances are updated until minimum distances are obtained. Break ties alphabetically.

a. Use Dijkstra's algorithm to determine the distances between vertex S and each other vertex. Clearly show the steps of the algorithm in the space provided.



← order in which I explored!

	Explored?	vertices	tentative distances
1	✓	S	0
2	✓	A	~~∞~~ 4
3	✓	B	~~∞~~ ~~6~~ 5
4	✓	C	~~∞~~ ~~12~~ 11
5	✓	D	~~∞~~ 14
6	✓	E	~~∞~~ 15

vertex	minimum distance to S
S	0
A	4
B	5
C	11
D	14
E	15

b. Which vertex is farthest away from S? E How far is it? 15

9. (8 points)

Give an example of a real world situation in which you would want to find:

a. a minimum weight Hamiltonian circuit.

Find the cheapest way to fly to several cities and return home.
vertices: cities
edge weights: cost of travel.

b. an Euler circuit.

Find the most efficient way to plow every road.
vertices: intersections of roads
edges: roads between intersections.

10. (Extra Credit: 5 points)

For each of the following, circle the correct answer. Write a few words to justify your answer.

True (False) (a) Given any connected finite graph with weighted edges, you can always find an optimal Hamiltonian circuit in a reasonable amount of time.

Brute force may take too long and it's our only guaranteed strategy.

(True) False (b) Given any graph where the degree of each vertex is even, you can always find an Euler circuit in a reasonable amount of time.

(This is supposing the graph is connected...) Yes. You can just wander around. Any missed edges lie on a loop that can be added.

(True) False (c) Given any connected graph, you can always find a minimal-weight spanning tree.

Kruskal's Algorithm always works and does so very quickly.

True (False) (d) The critical path algorithm always produces an optimal schedule.

No. We have examples from class where the decreasing time algorithm did better.

True (False) (e) Adding more processors will always give you a shorter schedule.

No. Once the total time per processor is below the critical time, additional processors cannot help