

Math 101-002
Exam 3 Makeup, Summer 2025

Name/CSU ID:

Be sure to read each question fully and carefully. There is space to the right of each multiple choice question to show work. If your work is correct, you can receive credit even with an incorrect choice.

Section 1: Networks and Spanning Trees (Ch. 7)

1. (12 points) Consider the following weighted network of six vertices. The weights represent distances in kilometers.

	A	B	C	D	E	F
A	-	7	9	-	-	14
B	7	-	10	15	-	-
C	9	10	-	11	-	2
D	-	15	11	-	6	-
E	-	-	-	6	-	9
F	14	-	2	-	9	-

- (a) (4 points) Draw the network as a graph, labeling all weights.
- (b) (4 points) Use Kruskal's Algorithm to find a minimum spanning tree. List the edges in the order selected.
- (c) (4 points) What is the total weight of the minimum spanning tree?

2. (4 points) What is a tree in graph theory? Why must Kruskal's algorithm always produce a tree? Explain in your own words.

3. (4 points) What is the difference between a spanning tree and a minimum spanning tree? Why do we care about finding the *minimum* one?

Section 2: Scheduling (Ch. 8)

4. (12 points) A project consists of the following task list and precedence relationships:

Task	Time (days)	Must follow
A	2	—
B	5	—
C	6	A
D	3	A, B
E	4	C
F	2	D, E

- (a) (4 points) Draw the project digraph with the processing times and precedence relations:

- (b) (4 points) Apply the Backflow Algorithm to find the critical time of each task. Draw the project digraph again if necessary.

- (c) (4 points) Use the Decreasing-Time Algorithm to schedule the above project on two processors. Assume alphabetical tie-breaks.

5. (4 points) Compare the Decreasing-Time Algorithm and the Critical-Path Algorithm. In what scenarios might one outperform the other?

6. (4 points) In your own words, explain what the purpose of a scheduling algorithm is. Why do we allow idle time in processor schedules, and how does that affect the overall completion time?

Discussion (10 points)

7. (10 points) You are helping a game development studio plan the final testing and launch of a video game. The project has tasks like coding the multiplayer engine, polishing the graphics, server testing, preparing launch trailers, and packaging for distribution. Some tasks must wait until others are completed. You have a team of testers and designers, and limited time before release day.

Describe how you could model the task dependencies as a directed graph. How would you identify the critical path? Which scheduling algorithm would you use to assign tasks to team members, and how would you deal with delays or last-minute changes?