University of Washington
Department of Computer Science and Engineering
CSE 417, Autumn 2020

October 27, 2020

Practice midterm, originally planned for November 2, 2020

NAME: ____________________ UW Net ID: ____________________

Instructions:

- Closed book, closed notes, no calculators
- Time limit: 50 minutes
- Answer the problems on the exam paper.
- If you need extra space use the back of a page
- Problems are not of equal difficulty, if you get stuck on a problem, move on.

1	/10
2	/10
3	/10
4	/10
5	/10
6	/15
7	/10
Total	/75

Problem 1 Graph Theory (10 points):

a) *True or false*: Let $G = (V, E)$ be an undirected graph. If G is a tree, then G is bipartite. Justify your answer[1].

b) *True or false*: Let $G = (V, E)$ be a directed graph with n vertices and m edges. It is possible to determine if G has a cycle in $O(n + m)$ time. Justify your answer.

[1] "Justify" means give a short and convincing explanation. Depending on the situation, justifications can involve counter examples, or cite results established in the text or in lecture.

Problem 2 Stable Matching (10 points):

Step through the Gale-Shapely stable matching algorithm on the instance below. (You may choose the proposals in any order.) The preference lists are:

$$M = \begin{bmatrix} m_1: & w_1 & w_2 & w_3 & w_4 \\ m_2: & w_1 & w_3 & w_4 & w_2 \\ m_3: & w_2 & w_1 & w_3 & w_4 \\ m_4: & w_2 & w_1 & w_3 & w_4 \end{bmatrix}$$

$$W = \begin{bmatrix} w_1: & m_3 & m_4 & m_1 & m_2 \\ w_2: & m_1 & m_2 & m_4 & m_3 \\ w_3: & m_3 & m_4 & m_1 & m_2 \\ w_4: & m_1 & m_2 & m_3 & m_4 \end{bmatrix}$$

Fill in the following table to trace the algorithm. The first two rows are given.

Round	Proposal	Result	Current Matching
0			$(m_1, *), (m_2, *), (m_3, *), (m_4, *)$
1	m_1 proposes to w_1	w_1 accepts m_1	$(m_1, w_1), (m_2, *), (m_3, *), (m_4, *)$
2	m_2 proposes to w_1	w_1 rejects m_2	$(m_1, w_1), (m_2, *), (m_3, *), (m_4, *)$
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Problem 3 Shortest Cycle (10 points):

Let $G = (V, E)$ be an undirected graph. Let $e = \{u, v\}$ be an edge in G. Give an $O(n + m)$ time algorithm that finds the shortest cycle in G which contains the edge e. Explain why your algorithm is correct.

Problem 4 Connected Components (10 points):

Suppose $G = (V, E)$ is an undirected graph with n vertices and n edges. (Note: G is not allowed to have self loops or parallel edges.)

a) What is the minimum number of connected components that G can have? Justify your answer.

b) What is the maximum number of connected components that G can have? Justify your answer.

Problem 5 Interval Scheduling (10 points):

The input for an interval scheduling problem is a set of intervals $I = \{i_1, \ldots, i_n\}$ where i_k has start time s_k, and finish time f_k. The problem is to find a set of non-overlapping intervals that satisfies a given criteria.

a) Suppose that you want to maximize the total length of the selected intervals. *True or false*: The greedy algorithm based on selecting intervals in order of decreasing length finds an optimal solution. Justify your answer.

b) Suppose that all intervals have the same length, and you want to maximize the total length of the selected intervals. *True or false*: The greedy algorithm based on selecting intervals in order of increasing start time finds an optimal solution. Justify your answer.

Problem 6 Recurrences (15 points):

Solve the following recurrences by unrolling the recursion tree. Express your answers as $O(f(n))$.

a)

$$T(n) = \begin{cases} 5T(\frac{n}{3}) + n & \text{if } n > 1 \\ 1 & \text{if } n \leq 1 \end{cases}$$

b)

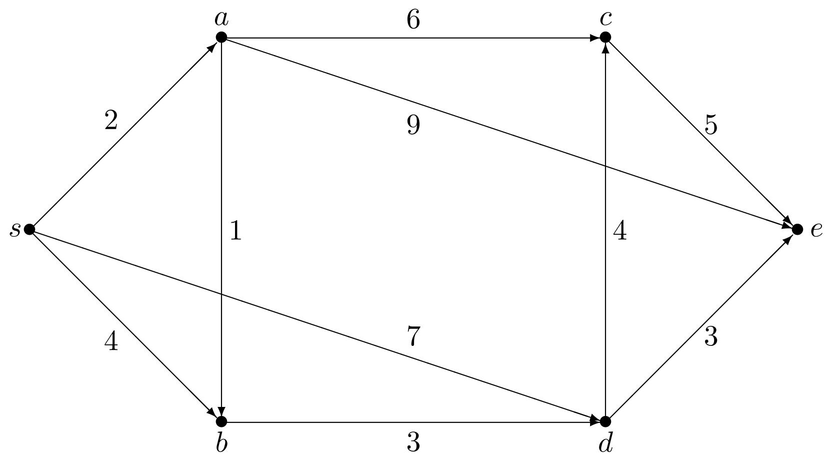
$$T(n) = \begin{cases} T(\frac{4n}{5}) + n & \text{if } n > 1 \\ 1 & \text{if } n \leq 1 \end{cases}$$

c)

$$T(n) = \begin{cases} 16T(\frac{n}{4}) + n^2 & \text{if } n > 1 \\ 1 & \text{if } n \leq 1 \end{cases}$$

Problem 7 Dijkstra's Algorithm (10 points):

Use the following graph to simulate versions of Dijkstra's algorithm in parts a) and c) starting from the vertex s.



a) Simulate Dijkstra's shortest path algorithm on the graph above by filling in the table. The entries should contain the preliminary distance values.

Round	Vertex	s	a	b	c	d	e
1							
2							
3							
4							
5							
6							

b) Draw the back edges found by your simulation of Dijkstra's algorithm.