University of Washington

Department of Computer Science and Engineering

CSEP 521, Spring 2011

April 26, 2011

Midterm Exam, Wednesday, April 27, 2011

NAME: ______________________________

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	/10
Total	/60

Problem 1 (10 points):

Consider the stable matching algorithm. Show how a w can be matched with all of the m's during the course of the algorithm. Hint: give the preference lists and describe an execution of the algorithm where some w is matched with each m in turn.

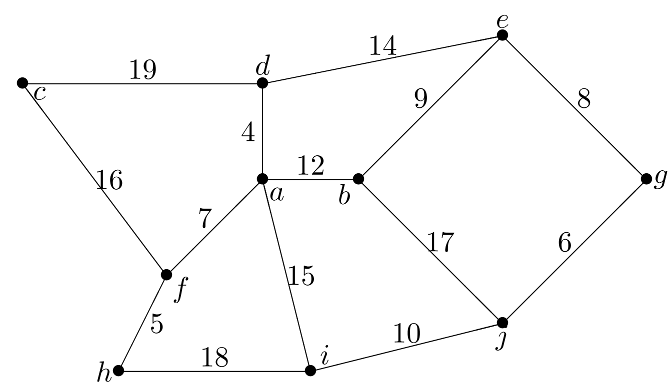
Problem 2 (10 points):

Let $G = (V, E)$ be an undirected graph, and v a vertex.

Give an $O(n + m)$ time algorithm that finds the shortest cycle in G that contains the vertext v.

Problem 3 (10 points):

Consider the following undirected graph G.



a) Highlight the edges of G that are in a minimum spanning tree.

b) Use the Edge Inclusion Lemma to argue that the edge (c, f) is in every Minimum Spanning Tree of G.

c) Use the Edge Exclusion Lemma to argue that the edge (b, j) is never in a Minimum Spanning Tree of G.

Problem 4 (10 points):

The binpacking problem is: Given a collection of items $I = \{i_1, \ldots, i_n\}$ where each item i_j has a s_j, an integer K, and a collection of bins $B = \{b_1, \ldots, b_m\}$ assign the items to bins such that the sum of the sizes of items assigned to each bin b_i is at most K. The goal is to minimize the number of bins that recieve items, i.e., to pack the items into as few bins as possible.

A greedy algorithm for the problem considers the items in order, and places each item in the first bin that has enough remaining space to hold the item.

Assume that the bin size is $K = 3$, and the items have sizes 1, 2, or 3.

a) Give an example that shows that the greedy algorithm does not necessary find the optimal solution (minimizes the number of bins).

b) Describe an algorithm which find an optimal solution. Explain why your algorithm minimizes the number of bins used. (You do not need to give a formal proof - just identify the key ideas.)

Problem 5 (10 points):

Give solutions to the following recurrences. Justify your answers.

a)

$$T(n) = \begin{cases} T(\frac{9n}{11}) + n & \text{if } n > 1 \\ 1 & \text{if } n \leq 1 \end{cases}$$

b)

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

Problem 6 (10 points):

Given an array of n real numbers, consider the problem of finding the maximum sum in any contiguous subvector of the input, for example, in the array

$$\{31, -41, 59, 26, -53, 58, 97, -93, -23, 84\}$$

the maximum is achieved by summing the third through seventh elements, where $59+26+(-53)+58+97 = 187$. When all numbers are positive, the entire array is the answer, while when all numbers are negative, the empty array maximizes the total at 0.

Give an $O(n \log n)$ divide and conquer algorithm for solving this problem.