University of Washington
Department of Computer Science and Engineering
CSE 421, Fall 2006

October 31, 2006

Midterm Exam, Friday, November 3, 2006

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
7	/10
Total	/70

Problem 1 (10 points):

Consider the stable matching problem.

a) Show that it is possible to have a *last-choice* match: There exists an instance of the problem with a stable matching M that has m matched with w, where w is m's last choice, and m is w's last choice.

b) Is it possible for a stable matching to have two *last-choice* matches: could a stable matching M have m_1 matched with w_1 where m_1 is w_1's last choice and w_1 is m_1's last choice, and m_2 matched with w_2 where m_2 is w_2's last choice and w_2 is m_2's last choice? Justify your answer.

Problem 2 (10 points):

Show that

$$\sum_{k=0}^{\log n} 4^n$$

is $O(n^2)$.

Problem 3 (10 points):

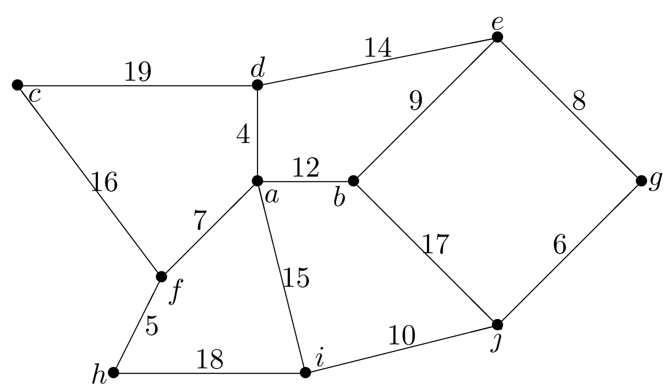
Let $G = (V, E)$ be an undirected graph.

a) True or false: If G is a tree, then G is bipartite. Justify your answer.

b) True or false: If G is not bipartite, then the shortest cycle in G has odd length. Justify your answer.

Problem 4 (10 points):

Consider the following undirected graph G.



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

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

Problem 5 (10 points):

The knapsack problem is: Given a collection of items $I = \{i_1, \ldots, i_n\}$ and an integer K where each item i_j has a weight w_j and a value v_j find a subset of the items which has weight at most K and maximizes the total value in the set. More formally, we want to find a subset $S \subseteq I$ such that $\sum_{i_k \in S} w_k \leq K$ and $\sum_{i_k \in S} v_k$ is as large as possible.
Suppose that the items are sorted in decreasing order of value, so that $v_i \geq v_{i+1}$. A simple greedy algorithm for the problem is:

$CurrWeight := 0;$
$Sack := \emptyset;$
for $j := 1$ **to** n
  if $CurrWeight + w_j \leq K$ **then**
    $Sack := Sack \cup \{i_j\}$
    $CurrWeight := CurrWeight + w_j$

a) Show that the greedy algorithm does not necessarily find the maximum value collection of items that can be placed in the knapsack.

b) Prove that if all weights are the same, then the greedy algorithm finds the maximum value set. (For convenience, you may assume that each item has weight 1).

Problem 6 (10 points):

Give solutions to the following recurrences. Justify your answers.

a)

$$T(n) = \begin{cases} 2T(\frac{n}{3}) + n & \text{if } n > 1 \\ 1 & \text{if } n \leq 1 \end{cases}$$

b)

$$T(n) = \begin{cases} 8T(\frac{n}{2}) + n^3 & \text{if } n > 1 \\ 0 & \text{if } n \leq 1 \end{cases}$$

Problem 7 (10 points):

A k-wise merge takes as input k sorted arrays, and constructs a single sorted array containing all of the elements of the input arrays.

a) Describe an efficient divide and conquer algorithm $\mathit{MultiMerge}(k, A_1, \ldots, A_k)$ which computes a k-wise merge of its input arrays.

b) What is the run time of your algorithm with input of k arrays of length n. Justify your answer.