Name: ______________________________

1	/14
2	/10
3	/10
4	/ 6
5	/ 7

CSE 421
Midterm Exam

February 15, 2002

Instructions:

- You have 50 minutes to complete the exam.
- Please do not turn the page until you are instructed to do so.
- Good luck!

1. (14 points, 2 each) Indicate for each of the following if it is **true or false** by circling the appropriate answer. (Remember, a statement is true only if it is true in all instances.)

   - True or False: There is an algorithm to determine whether a given undirected graph on n nodes is a tree that runs in $O(n)$ time.
     True

   - True or False: Suppose that G is a connected, undirected graph. If removing edge e from G disconnects the graph, then e is a tree edge in the depth-first search of G.
     True

   - True or False: Let $G = (V, E)$ be a directed graph, where $(u, v) \in E$. If in a depth-first search on G the edge (u, v) is a cross edge, then u is visited before v.
     False

   - True or False: Suppose that G is a directed, acyclic graph and has a topological ordering with v_1 the first node in the ordering and v_n the last node in the ordering. Then there is a path in G from v_1 to v_n.
     False

   - True or False: It is possible for a directed graph G with exactly two topological orderings to have exactly one node of in-degree 0.
     True

2. (10 points, 2 each) For the following problems, let G be an undirected, weighted graph that is connected. Please indicate whether each statement is **true or false** by circling the appropriate answer.

   - True or False: Suppose that e is a minimum weight edge G, and all the edge weights are distinct. Then e is always contained in the minimum spanning tree of G.
     True

   - True or False: The path between a pair of vertices in a minimum spanning tree for G must be a shortest (*i.e.* least-cost) path between the two vertices in G.
     False

   - True or False: Suppose that G has a unique minimum spanning tree. Then all edges have distinct weights.
     False

- True or False: Say edge e is the cheapest edge in some cycle of G and all edge weights are distinct. Then e is in the minimum spanning tree of G.

  False

- True of False: Suppose we run Dijsktra's algorithm on G starting at some vertex, s. Recall that at each step of the algorithm, we add another vertex to the set S. Suppose we add vertex u to S before we add the vertex v to S. Then the distance from s to u in G is no greater than the distance from s to v.

  True

3. (10 points) Consider the generalization of the stable matching problem, in which certain man-woman pairs are explicitly *forbidden*. Specifically, we have a set M of n men, a set W of n women, and a set $F \subseteq M \times W$ of pairs who are simply *not allowed* to get married. Each man m ranks all the women w for which $(m, w) \notin F$, and each woman w' ranks all the men m' for which $(m', w') \notin F$. Note that under these more general definitions, a stable matching need not be a perfect matching.

   Consider the following algorithm for finding a stable matching that consists only of unforbidden pairs.

   > Initially all $m \in M$ and $w \in W$ are free (i.e., $S = \emptyset$) While there is a man m who is free and hasn't proposed to every woman w for which $(m, w) \notin F$ Choose such a man m Let w be the highest-ranked woman in m's preference list to which m has not yet proposed If w is free then (m, w) become engaged Else w is currently engaged to m' If w prefers m' to m then m remains free Else w prefers m to m' (m, w) become engaged m' becomes free Endif Endif Endwhile Return the set S of engaged pairs

   (2 points each) For each of the following statements about this algorithm, indicate whether or not the statement is **true or false**, by circling the appropriate answer.

   - True or False: Any woman w remains engaged from the point at which she receives her first proposal, and the sequence of partners to which she is engaged get better and better (in terms of her preference list).

     True

   - True or False: If a man m is free at the end of the algorithm, then he must have proposed to every non-forbidden woman.

     True

   - True or False: If a woman w is free at the end of the algorithm, then it must be that no man ever proposed to w.

     True

- True or False: At the end of the algorithm, there can be a man m and a woman w, such that $(m, w) \notin F$, but neither of which is part of any pair in the matching S.

  False

- True or False: At the end of the algorithm, there can be a pair $(m, w) \in S$ and a man m' that is free, $(m', w) \notin \mathcal{F}$, but such that w prefers m' to m.

  False

4. (6 points) **Fill in the blanks**:

   - (3 points) Give an example of preference lists for two men and two women such that the male-optimal stable matching (where each man is paired with his best valid partner) and the female-optimal stable matching (where each woman is paired with her best valid partner) are different.

     man 1's preference list: $(w1, w2)$

     man 2's preference list: $(w2, w1)$

     woman 1's preference list: $(m2, m1)$

     woman 2's preference list: $(m1, m2)$

   - (3 points) Suppose that e is the maximum weight edge in an undirected, weighted graph G, where the weights on all edges are distinct. Give a necessary and sufficient condition for e to be in the minimum spanning tree of G.

     Necessary and sufficient condition: Removing e disconnects the graph.

   - Consider a divide and conquer algorithm where the running time $T(n)$ satisfies

     $$T(n) = 5T(n/2) + n$$

     for $n \geq 1$ and $T(1) = 0$. The running time of the algorithm is $O(n^{log_2 5})$

5. (7 points) Consider the following problem. You are given two sequences over some alphabet: sequence S consists of $s_1, \ldots, s_n$ and sequence S' consists of $s'_1, \ldots, s'_m$. Your goal is to determine whether S' is a subsequence of S (i.e., S' can be obtained from S by deleting $n - m$ of the letters in S.) For example, $S' = abbc$ is a subsequence of $S = adbbc$, but $S' = abcc$ is not.

   Now consider the following greedy algorithm for this problem. Find the first letter in S that is the same as s'_1, match these two letters, then find the first letter after this that is the same as s'_2, and so on. We will use $k_1, k_2, \ldots$ to denote the positions of the match have we found so far, i to denote the current position in S, and j the current position in S'. We obtain the following pseudocode for this algorithm:

   Initially $i = j = 1$ While $i \leq n$ and $j \leq m$ If s_i is the same as s'_j, then let $k_j = i$ (so that $s_{k_j} = s'_j$) let $i = i + 1$ and $j = j + 1$ otherwise let $i = i + 1$

EndWhile If $j = m + 1$ return the subsequence found: positions $k_1, \ldots k_m$.
Else return that "S' is not a subsequence of S"

For each of the following, fill in the blanks.

- (2 points) The running time of this algorithm is **O(n)**.
- (5 points) Suppose that there is a match, i.e. S' is the same as the subsequence at positions $l_1, \ldots, l_m$ of S. (Note that there could be other subsequences of S that also give S'.) We'd like to prove that in this case, the greedy algorithm is guaranteed to find a match. Fill in the statement that you would prove by induction in order to show this. (You do not need to supply a proof of the statement.)
  $\forall j, 1 \leq j \leq m, k_j \leq l_j$ and $S_{k_j} = S'_j$

6. 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.

Solution: Let us assume that $maxSum(1, n)$ returns the required optimal solution. We split the array into two halves and use $maxSum(1, \frac{n}{2})$, $maxSum(\frac{n}{2} + 1, n)$ recursively to compute the solutions s_1,s_2 respectively to each of the two subproblems.

Now for the merge operation, we need to consider all subvectors which cross the boundary between the two subvectors $[1, \frac{n}{2}]$ and $[\frac{n}{2} + 1, n]$. To do this, we find the maximum sum subvector v_1 which ends at the index $\frac{n}{2}$ and starts from some element of the subvector $[1, \frac{n}{2}]$. Let that subvector be $[i, \frac{n}{2}]$. v_1 can be computed in $O(n)$ time in one single pass over the $[1, \frac{n}{2}]$. We similarly find the subvector $v_2 = [\frac{n}{2} + 1, j]$ which maximizes the sum in the other subvector starting from $\frac{n}{2} + 1$. Compute the sum s_3 of the elements in the subvector $[i, j]$.

Return maximum among the numbers s_1,s_2 and s_3.

Time Complexity: We see that $T(n) = 2T(\frac{n}{2}) + O(n)$. Hence the algorithm is $O(nlogn)$.