CS325: Analysis of Algorithms, Fall 2017

Final Exam

- *I don't know policy:* you may write “I don't know” and nothing else to answer a question and receive 25 percent of the total points for that problem whereas a completely wrong answer will receive zero.
- There are 8 problems in this exam.

<table>
<thead>
<tr>
<th>Problem 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 2</td>
<td></td>
</tr>
<tr>
<td>Problem 3</td>
<td></td>
</tr>
<tr>
<td>Problem 4</td>
<td></td>
</tr>
<tr>
<td>Problem 5</td>
<td></td>
</tr>
<tr>
<td>Problem 6</td>
<td></td>
</tr>
<tr>
<td>Problem 7</td>
<td></td>
</tr>
<tr>
<td>Problem 8</td>
<td></td>
</tr>
</tbody>
</table>
**Problem 1.** [3 pts] For each of the following minimum spanning tree algorithms, mark the first *four* edges that the algorithm would find when it is performed on the given graph.

(a) Borvka:

(b) Jurník, when it starts from the vertex in the center:

(c) Kruskal:

**Problem 2.** [4 pts] Consider the three minimum spanning tree algorithms (Borvka, Jurník, and Kruskal) as presented in class or in lecture notes. Our proofs of correctness relied on the assumption that the edge weights are distinct.

- Respond True or False for each of the following three statements.
- For each of the *False statements*, justify your answer by giving an example of a graph with non-distinct edge weights that the algorithm fails to find its MST.

(a) Borvka correctly finds the MST of a graph with non-distinct edge weights.
   - F
   - T

(b) Jurník correctly finds the MST of a graph with non-distinct edge weights.
   - F
   - T

(c) Kruskal correctly finds the MST of a graph with non-distinct edge weights.
   - T
Problem 3. [3 pts] Let $G = (V, E)$ be a connected graph with $n$ vertices, and $n + 1$ edges.

(a) Suppose the weight of every edge of $G$ is 10. What is the weight of the minimum spanning tree of $G$?

(b) Now, assume that $G$ has exactly one edge of weight 20, exactly one edge of weight 5, and the weight of every other edge is 10. What are the possibilities for the weight of the MST of $G$?

Justify your responses to both (a) and (b).

(a) $10(n-1)$, since MST contains $n-1$ edges

(b) 2 possibilities: $5 + 10(n-2) \quad 5 + 20 + 10(n-3)$

all MST's will contain 5. (why?)

Problem 4. [4 pts] Which of the following job scheduling algorithms always construct an optimal solution? Mark an algorithm True if it always constructs an optimal solution, and False, otherwise. Provide counterexamples for those that you mark False.

(a) Choose a job $x$ that conflicts with the fewest other jobs, discard all jobs that conflict with $x$, and recurse. $F$

(b) Choose the job $x$ that starts last, discard all jobs that conflict with $x$, and recurse. $T$

(c) Choose the job $x$ with shortest duration, discard all jobs that conflict with $x$, and recurse. $F$

(d) If no jobs conflict, choose them all. Otherwise, discard the job with longest duration and recurse. $F$

\[ a) \]
Problem 5. [3 pts] What is an optimal Huffman code for the set of frequencies in each of the following cases (specify the codes, or draw the corresponding binary trees)?

(a) $a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13$.
(b) $a : 10, b : 10, c : 10, d : 10, e : 10, f : 10, g : 10, h : 10$.

Problem 6. [4 pts] For each of the following trees, determine if it can be the optimal Huffman tree for any set of frequencies. For those that can be, find a set of frequencies for which the tree is the optimal Huffman tree, and show the correspondence between the frequencies and the leaves of the tree.

(a) \( \begin{array}{c}
\text{2} \\
\text{1 1}
\end{array} \)
(b) \( \begin{array}{c}
\text{2 2 3 3}
\end{array} \)
(c) \( \begin{array}{c}
\text{X}
\end{array} \)
Problem 7. [6 pts] For each of the following statements, respond True, False, or Unknown.

(a) If a problem is decidable then it is in P.  \( F \)
(b) For any decision problem there exists an algorithm with exponential running time.  \( F \)
(c) \( P = NP \).  \( U \)
(d) All NP-complete problems can be solved in polynomial time.  \( U \)
(e) If there is a reduction from a problem \( A \) to \text{CIRCUIT SAT} then \( A \) is NP-hard.  \( F \)
(f) If problem \( A \) can be solved in polynomial time then \( A \) is in NP.  \( T \)
(g) If problem \( A \) is in NP then it is NP-complete.  \( F \)
(h) If problem \( A \) is in NP then there is no polynomial time algorithm for solving \( A \).  \( U \)

Problem 8. [4 pts] A \( k \)-CNF formula is a conjunction (AND) of a set of clauses, where each clause is a disjunction (OR) of a set of exactly \( k \) literals. For example,

\[(a \lor b \lor c \lor \neg d \lor \neg e) \land (\neg a \lor b \lor c \lor \neg x \lor \neg y) \land (\neg x \lor y \lor c \lor d \lor a)\]

is a 5-CN. The \( k \)-SAT problem asks if a \( k \)-CNF formula is satisfiable. In class we saw that 3-SAT is NP-hard. In contrast, 2-SAT is polynomially solvable, as it is mentioned in GA4.

(a) Show that 4-SAT is NP-Complete (For partial credit, specify all the statements you need for showing that 4-SAT is NP-complete, even if you cannot prove them).

(b) Describe a polynomial time algorithm to solve 1-SAT.