Johan Karlander, KTH, CSC

Teoritenta i Algoritmer (datastrukturer) och komplexitet för KTH DD1352–2352 2014-06-05, klockan 9.00–12.00

No aids are allowed. 12 points are required for grade E, 15 points for grade D and 18 points for grade C.

If you have done the labs you can get up to 4 bonus points. If you have got bonus points, please indicate it in your solutions.

In all solutions you can assume that $P \neq NP$.

1. (8 p)

   Are these statements true or false? For each sub-task a correct answer gives 1 point and an answer with convincing justification gives 2 points.

   a. The problem of deciding if a graph is connected is an NP-Complete problem.

   b. If a Divide and Conquer-algorithm has a time complexity $T(n)$ given by the recursion formula

   $$T(n) = 2T(\frac{n}{4}) + cn$$

   then $T(n) \in O(n)$.

   c. If a problem A can be reduced to a problem B in polynomial time and we know that B is NP-hard, then we know that A is NP-Complete.

   d. The problem of deciding if a Turing machine M halts on all inputs is an undecidable problem.

2. (3 p)

   As you know, Dijkstra's algorithm is an algorithm for finding shortest paths in weighted directed graphs. It works for graphs with positive weights, but not always for graphs with negative weights. Explain how Dijkstra's algorithm works and then explain why it can not handle negative weights.

3. (3 p)

   From the lectures you know that the max Max Flow Algorithm can be used to solve the matching problem. Now we look at a similar problem:

   Let G be a directed graph, and $A \subseteq V(G), B \subseteq V(G)$ two sets of nodes such that $A \cap B = \emptyset$. We want to find a maximum size set M of edge-disjoint paths from A to B. This means that all paths in M starts in a node in A and ends in a node in B. No paths in M have any common edges (but might have common nodes). Explain why this problem can be seen as a generalization of the matching problem. Show how we can use the Max Flow Algorithm to find such a set M.

4. (3 p)

   We will here look at a suggested reduction from SUBSET SUM to PARTITIONING. Let $a_1, a_2, ..., a_n$ be positive integers and K a positive integer. We want to know if there is a subset sum equal to K. We construct an instance of PARTITIONING as $a_1, a_2, ..., a_n, X$. Decide what X should be. Prove that the reduction is correct.

5. (3 p)

   This problem is about TSP (Traveling Salesperson Problem). we have a complete, undirected graph G with edge weights d_{ij} where d_{ij} is the weight (length) of the edge (i, j). We want to find a shortest Hamiltonian cycle in the graph.

   We can actually solve this problem recursively. Let $S \subseteq \{1, , ..., n\} = V(G)$ such that $1 \in S$, let $i \in S$ and let $W(S, i)$ be the length of the shortest possible *path* visiting each node in S exactly once (and no nodes outside S), starting at 1 and ending in i.

   We set $W(\{1\}, 1) = 0$ and $W(S, 1) = \infty$ for all $|S| > 1$. Show that for all $j \neq 1$ we have

   $$W(S, j) = \min_{i \in S, i \neq j} W(S - \{j\}, i) + d_{ij}.$$

   Show how we can use this recursion formula to find an algorithm that solves TSP, i.e., finds the length of a shortest Hamiltonian Cycle. What is the time complexity of your algorithm if n is the input size? (Don't expect to find an *efficient* algorithm.)