TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

TU Delft

Exam IN4301 Advanced Algorithms – Part I

November 4, 2013, 14:00–16:00

- **Please answer questions 1,2, and 3 on a separate sheet from questions 4, 5, and 6.**
- This is a closed book examination with 6 questions worth of 50 points in total.
- Your mark for this exam part will be the number of points divided by 5.
- If you have at least a 5.0 for this exam part as well as for the second part (January), the average of the two programming exercises, as well as the average of the homework exercises, your final mark for this course is the average of these three marks, rounded to the nearest half of a whole number. That is, 9.7 is rounded to 9.5, and 5.8 is rounded to 6.
- Use of book, readers, notes, and slides is not allowed.
- Use of (graphical) calculators is not permitted.
- Specify your name, student number and degree program, and indicate the total number of submitted pages on the first page.
- Write clearly, use correct English, and avoid verbose explanations. Giving irrelevant information may lead to a reduction in your score.
  Notice that almost all questions can be answered in a few lines!
- This exam covers Chapters 10 and 11 of Kleinberg, J. and Tardos, E. (2005), *Algorithm Design*, all information on the slides of the course of the first 7 lectures, and the set of papers as described in the study guide.
- The total number of pages of this exam is 2 (excluding this front page).

1. Consider the following problem (3-SAT). Given a set of logical (boolean) variables $X = \{x_1, x_2, \ldots, x_n\}$, and a Boolean formula in 3-CNF (i.e., each clause is a disjunction with at most 3 literals) with a set C of m clauses (with $m \leq n^3$). Decide whether there exists an assignment to the variables such that all clauses in C are satisfied.
   (a) (6 points) Describe a search tree algorithm for this problem with an upper bound on the run time of strictly less than $o(2^n)$.
   (b) (6 points) Give a tight upper bound on the run time complexity of the search tree algorithm you provided and explain how you arrive at this.

2. Given is one machine, a set S of jobs, each $j \in S$ with a length p_j and a weight w_j, and a directed acyclic graph (S, E) of jobs representing precedence constraints. The problem is to find a schedule with completion time C_j for each job j obeying the precedence constraints, and with a minimum sum of weighted completion times, i.e., minimize $\sum_{j \in S} w_j C_j$.

   The following function defines the value of the optimal solution for this scheduling problem:

   $M[\emptyset] \leftarrow 0$
   for *each subset* $S' \subseteq S$ *in increasing size* **do**
   $\quad \mathrm{LAST}(S') \leftarrow \{j \in S' \mid$ there is no job $i \in S'$ with $(j, i) \in E\}$
   $\quad M[S'] \leftarrow \ldots$
   return $M[S]$

   (a) (4 points) Provide the missing statement (what should be on the dots?).
   (b) (3 points) Give an analysis of a tight upper bound on the run time of this algorithm.

3. Give arguments for each of the following:
   (a) (3 points) If $H = (V', E')$ is a subgraph of G (so $V' \subseteq V$ and $E' \subseteq E$) then the treewidth of H is less than or equal to the treewidth of G.
   (b) (3 points) If $G = (V, E)$ has two unconnected components A and B such that $A \cup B = V$ then the treewidth of G is equal to the maximum of the treewidths of the subgraphs induced by A and B, respectively (i.e., $tw(G) = \max\{tw(A), tw(B)\}$).

Please turn this page to continue.

Please start with a fresh sheet.

4. Consider the following optimization version OPTPART of PARTITION: Given a set X of positive integers, partition X into disjoint subsets A and B of X such that $\max\{\Sigma_{x\in A}\, x, \Sigma_{y\in B}\, y\}$ is as small as possible. This problem is an NP-hard problem. Consider the following approximation algorithm for OPTPART:

   ```
   Sort X in increasing order;
   a := 0; b := 0;
   for i = 1 to n do
       if a < b then
           a := a + X[i]
       else
           b := b + X[i]
   return max{a, b};
   ```

   (a) (5 points) Apply this approximation algorithm to the instances $X_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and $X_2 = \{40, 40, 80\}$ and compute the performance ratio of the algorithm for both instances.

   (b) (5 points) Show that the absolute difference between a and b for an arbitrary set X is bounded above by $\max_{x\in X}\{x\}$.

   (c) (7 points) Give a constant c such that the approximation ratio of this algorithm is bounded above by c. Motivate your answer.

5. (4 points) Suppose you know some NP-hard decision problem X and a minimisation problem A. As it turns out, there is a polynomial function f that maps yes-instances of X to instances of problem A where the value of the optimum solution is at most 10, while no-instances of X are mapped to instances of A having an optimum value of 12 or more.

   What can you conclude w.r.t. the possible values of the approximation threshold for A, assuming that $P \neq NP$?

6. (4 points) If we construct an approximation algorithm ALG for a problem A, we often need so-called *tight* examples. Give a definition of a tight example and argue why we need tight examples for approximation algorithms.

End of test