TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

Exam IN4301 Advanced Algorithms – Part I

November 3, 2014, 14:00–16:00

- Please answer questions 1,2, 3, and 4 on a separate sheet from questions 5, 6, and 7.
- This is a closed book examination with 7 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 problem of deciding whether there exists a VERTEX COVER of size at most k in a given undirected graph G. The most efficient bounded search tree algorithm for this problem dealt with in the lecture is the following:

   1. If G has no edges, return true.
   2. Else if G has more than kn edges, return false.
   3. Else if G has a vertex v with degree 1, put its neighbor u in the cover and solve the problem for $G - \{u, v\}$ and $k - 1$.
   4. Else if G has a vertex v with degree 2, ...
   5. Else if G has a vertex v with degree 3 or more, return true if one of the following two subproblems returns true:
      (a) Put v in the cover and solve the problem for $G - \{v\}$ and $k - 1$.
      (b) Put all neighbors $N(v)$ of v in the cover and solve the problem for $G - N(v)$ and $k - |N(v)|$.

   (a) (3 points) Complete the missing case (item 4, degree 2). (Hint: The worst-case bound on the size of the subproblems should be better than the bound for the last case.)

   (b) (4 points) Give the argument why your answer to the previous question is correct.

2. (5 points) Let $(Tr = (T, F), \{Vt : t \in T\})$ be a tree decomposition of $G = (V, E)$ and let $t \in T$. Remove t from T and V_t from G. The result is a set of independent trees $T_1, \ldots, T_d$. Let G_{T_i} be the resulting subgraphs associated with the trees T_i. Prove that these subgraphs are not connected, in other words: show that there exists no edge (u, v) with u in G_{T_i} and v in G_{T_j} for $i \neq j$.

3. 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 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: $\text{OPT}(\emptyset) = 0$ and otherwise $\text{OPT}(S) = \min_{j \in \text{LAST}(S)}\{\text{OPT}(S - \{j\}) + w_j p(S)\}$, where $\text{LAST}(S)$ is set of jobs in S that do not need to precede others, and $p(S) = \sum_{i \in S} p_i$.

   (a) (3 points) Give an analysis of a tight upper bound on the run time of a recursive implementation of this algorithm without memoization.

   (b) (3 points) Give an analysis of a tight upper bound on the run time of an iterative implementation of this algorithm using dynamic programming.

   (c) (2 points) Explain in which order the values should be computed in the iterative implementation.

4. Consider the problem of deciding whether a vertex cover exist of size at most k in a given undirected graph G. One of the rules to arrive at a small kernel for such problems uses so-called *crown structures*.

   (a) (3 points) Given a graph $G = (V, E)$, provide the definition of a crown structure.

   (b) (2 points) Given a vertex cover problem for G and a crown structure as defined in your previous answer, what is the reduced problem?

Please turn this page to continue.

Please start with a fresh sheet.

5. We consider the problem INDEPENDENT SET: given an undirected graph $G = (V, E)$ and a positive integer K, does there exist a subset $V' \subseteq V$ with at least k nodes such that for every two elements $u, v \in V'$ it holds that $\{u, v\} \notin E$? It is known that this problem is NP-hard. We are considering the maximization problem INDEPENDENT SET-B, where the problem is to find an independent set of maximum cardinality when the degree of every vertex $v \in V$ is limited to a constant B. This problem is NP-hard, too. In this question, you are requested to show that INDEPENDENT SET-B is in APX.
   (a) (3 points) Give an approximation algorithm for INDEPENDENT SET-B based on the following idea: every time you add a node $v \in V$ to the independent set, you remove v with all its neighbours from V.
   (b) (5 points) A subset W of V is a dominating set of G if for all $v \in V$ it holds that either $v \in W$ or there is an edge $u, v \in E$ such that $u \in W$. Show that the set V' given by the approximation algorithm is an independent set and also a dominating set.
   (c) (5 points) Show that the optimal independent set V_{opt} cannot be more than B times larger than the set V' given by the approximation algorithm. Hint: use the fact that the set V' returned by the algorithm is a dominating set!

6. (6 points) Let A be a minimisation problem. It is known that the problem to decide for an arbitrary instance x of A whether $opt(x) \leq 5$ or $opt(x) > 8$ is NP-hard. For which values of c can you prove that there does not exist a c-approximation algorithm for A, unless P=NP?

7. (6 points) Give a clear and concise description of the relationships between (i) a tight example, (ii) an approximation ratio and (iii) an approximation threshold.

End of test