Utrecht University
Faculty of Science
Department of Information and Computing Sciences

Final Exam Algorithms for Decision Support

- Switch off your smart phone, PDA and any other mobile device and put it far away.
- If you want to go to the bathroom, you have to hand in your mobile phone.
- This exam consists of **4** questions.
- Answers should be provided in English.
- All your answers should be clearly written down and provide a clear explanation. Unreadable or unclear answers may be judged as false.
- Please write down your name and student number on every exam paper that you hand in.

Question 1: Reading and Filling *(2 + 2 + 2 + 2 + 2 = 10 points)*

In this assignment, you fill in the blanks for an NP-hardness proof.

There is an alternative definition of the BinPacking problem: Given a set $S = \{a_1, \cdots, a_n\}$ of n items, where the i-th item has integral size $1 \le a_i \le B$. The problem is whether it is possible to pack the items in S into at most k bins, where each bin has capacity B. I.e., for each bin, the total size of all items in this bin is at most B; each item is placed in exactly one bin.

The following is an NP-hardness proof with some parts missing.

Proof. We prove that ____**(a)**____ is NP-complete by a ____**(b)**____ from ____**(c)**____. For any instance of BinPacking, (S, k), where $S = \{a_1, a_2, \cdots, a_n\}$, we transform it to an instance of MachineMinimization as follows:

- There are n jobs.
- Job J_i has processing time a_i.
- For any job J_i, its feasible interval $I_i = [0, B]$.
- The job can or cannot be scheduled feasibly on $m = k$ machines.

This transformation can be done in polynomial time.

Now, we show that ____**(d)**____. If there is a packing of S, the elements in S can be partitioned into at most k parties such that the total size of each part is at most B. Consider the jobs corresponding to the items of one of the $m = k$ parties. The total processing time of these jobs is at most B. Since the jobs have uniform feasible intervals $[0, B]$, they can be feasibly scheduled on one machine.

Next, we show that ____**(e)**____. If there is a feasible schedule of the instance of MachineMinimization problem, the jobs can be scheduled on $m = k$ machines. Since each feasible interval is uniformly $[0, B]$, the total processing time of jobs on one machine is at most B. Hence, the sum of size of items that are corresponding to the jobs assigned to one machine is at most B. □

Fill the blanks using the following elements (you only need to write down the indices $1, 2, 3, ...$ of the elements). Note that not every element should appear in the proof.

(1) MachineMinimization

(2) BinPacking

(3) exponential-time reduction

(4) polynomial-time reduction

(5) if there is no feasible schedule of the MachineMinimaization problem instance using at most m machines, the corresponding instance of BinPacking cannot be packed using k bins.

(6) if there is a feasible schedule of the MachineMinimaization problem instance using at most m machines, the corresponding instance of BinPacking has a packing with at most k bins.

Question 2: TRUE or FALSE and MS *(2 + 2 + 2 + 2 + 2 = 10 points)*

For the following subquestions, indicate **TRUE** or **FALSE**. To get the full marks, you have to **fully explain your FALSE answer**. You don't need to give any further explanation for the TRUE answer.

(a) Any unrecognizable language is undecidable.

(b) Any problem in NP is Turing-decidable.

(c) Assume that we reduce problem A to problem B. If we can solve A, we can solve B.

(d) If someone shows that an NP-complete problem is polynomial-time solvable, then P = NP.

(e) Multiple choice. In the guest lecture, professor Dennis Huisman described amongst others methods to solve Rolling Stock planning with more than one type of train units. What methods are used to solve this problem? (Write down the letter of the answer. No explanation is needed.)

(1) Graph coloring and a composition graph.

(2) Integer Linear Programming and unit sets.

(3) Multicommodity flow and a transition graph.

(4) Shortest path algorithms and a track graph.

Question 3: Find the smallest square *(15 points)*

Consider the following SquareFinding problem. Given a list of n rectangles $r_1, r_2, \cdots, r_n$, the rectangle r_i has width w_i and height h_i which are integers for each $1 \leq i \leq n$. The goal is to pack the items into a smallest square by shifting the rectangles (see Figure 1, notice that rotation is not allowed).

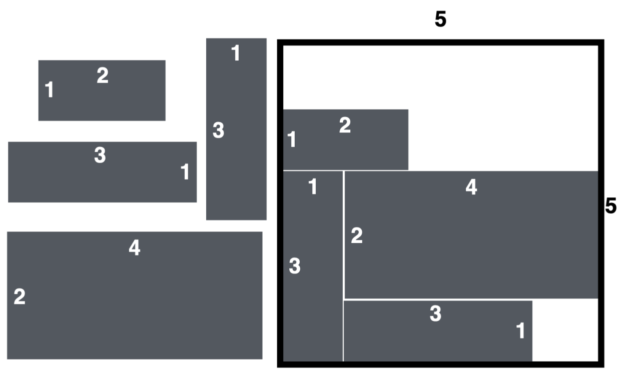

Figure 1: Illustration

(a) Define the decision version of the SquareFinding problem.

(b) Show that the decision version of SquareFinding is NP-complete.

(You can use 7 points to buy the hint for this question.)

Question 4: Connect terminals *(15 points)*

Given a graph $G = (V, E)$ and a set of terminals $T \subseteq V$, the problem SteinerTree aims at finding a minimum subset of edges $E' \subseteq E$ to connect all terminals T.

(a) Define the decision version of the SteinerTree problem.

(b) Use VertexCover to show that the decision version of SteinerTree is NP-complete.

(You can use 7 points to buy the hint for this question.)