Due: 31st of October 2016 at 11:59pm

COMP 2007 – Assignment 5

All submitted work must be done individually without consulting someone elses solutions in accordance with the Universitys Academic Dishonesty and Plagiarism policies.

Answers to questions 1a(i–iii), 1b(i–iii), 1c and 2 should be submitted via Blackboard as pdf (no handwriting!). Answers to questions 1a(iv) and 1b(iv) should be submitted via Ed.

Questions

This assignment is all about polynomial time reductions and NP-completeness. To prepare for it I suggest you read Chapter 8.1-4 in detail, and Chapters 7.5-12 might also be useful to see how correctness of a reduction is proven.

1. **[85 points]** Consider the following four problems:

   - **The Knapsack problem:** You are given a set $U = \{u_1, \ldots, u_{n_1}\}$ of n_1 items, and each item u_i has a weight $w_i > 0$, and a value $v_i > 0$. You are also given a weight capacity value $W > 0$, and a target value $V > 0$. You would like to decide if there exists a subset of the items, with weight less than or equal to W and total value greater than or equal to V.
   - **The Fair Split problem:** You are given a set $S = \{s_1, \ldots, s_{n_2}\}$ of n_2 items, and each item s_i has a value $v_i > 0$. You would like to decide if it is possible to split F into two subsets in a fair way: The total value of each of the two subsets must be exactly equal.
   - **The Exact Fill problem:** You are given a set $F = \{f_1, \ldots, f_{n_3}\}$ of n_3 items, and each item f_i has a weight $w_i > 0$. You are also given a weight capacity value $C > 0$. You would like to decide if there is a subset of the items, whose total weight is exactly equal to the given capacity.
   - **The One-of-Everything problem:** You are given a set of items $X = \{x_1, \ldots, x_m\}$ and a collection of subsets $S = \{S_1, \ldots, S_k\}$ of X, that is, $S_i \subseteq X$. You would like to decide if there is set of subsets of S whose union is X and such that no item appears twice (you want exactly one of every item in X).

   All problems are closely related. Assume you are given an algorithm for the Knapsack *decision* problem mentioned above. Assume that this is given to you as a black box (**you may not modify it!**). But you do know that its running time is $f(n_1, W, V)$, where n, W and V are three parameters of the knapsack instance.

   (a) Design an algorithm that solves the Fair Split problem, using the decision version of the Knapsack problem as a sub-procedure. Lower complexity generally gives higher marks. (Note that the question specifically asks you to use Knapsack as a sub-procedure. Using a different sub-procedure for your reduction will give 0 points.)

      i. Describe your reduction.
      ii. What is the running time of your algorithm?
      iii. Prove the correctness of your algorithm.
      iv. Implement your algorithm. Instructions for the implementation part is available on Ed.

(b) Design an algorithm that solves the Exact Fill problem, using the Fair Split problem as a sub-procedure. Lower complexity generally gives higher marks. (Note that the question specifically asks you to use Fair Split as a sub-procedure. Using a different sub-procedure for your reduction will give 0 points.)

i. Describe your reduction.

ii. What is the running time of your algorithm?

iii. Prove the correctness of your algorithm.

iv. Implement your algorithm, using your implementation for the Fair Split algorithm as a sub-procedure. Instructions for the implementation part is available on Ed.

(c) Design an algorithm that solves the One-of-Everything problem, using the Exact Fill problem as a sub-procedure.

You are given the reduction below, and your task is to prove the correctness of the reduction. For $1 \leq i \leq m$ and $1 \leq j \leq k$, let $e_{ji} = 1$ if $u_i \in S_j$ and 0 if $u_i \notin S_j$. Furthermore, let

$$w_j = \sum_{i=1}^{m} e_{ji}(k+1)^{i-1}$$

and

$$C = \sum_{i=0}^{m-1} (k+1)^i.$$

Call an algorithm for Exact Fill using the parameters $w_1, \ldots, w_k$ and C.

2. **[15 points]** Assume that the One-of-Everything problem is NP-complete.

(a) Prove that the Exact Fill problem is NP-complete.

(b) Prove that the Fair Split problem is NP-complete.

(c) Prove that the Knapsack problem is NP-complete.