TECHNICAL UNIVERSITY OF DENMARK

Written examination, date: XX.XX.XXXX

Course name: Computationally Hard Problems

Course number: 02249

Aids allowed: All written aids are allowed

Exam duration: 4 hours

Weighting:
Exercise 1 – 10 %
Exercise 2 – 15 %
Exercise 3 – 15 %
Exercise 4 – 20 %
Exercise 5 – 10 %
Exercise 6 – 10 %
Exercise 7 – 20 %

TECHNICAL UNIVERSITY OF DENMARK Written test, XX.XX.XXXX

Course: Computationally Hard Problems, Course No. 02249

Aids allowed: All written aids are allowed.

Weighting: Exercise 1 – 10%, Exercise 2 – 15%, Exercise 3 – 15%,

Exercise 4 – 20%, Exercise 5 – 10%, Exercise 6 – 10%, Exercise 7 – 20%.

Exercise 1: An integer program (IP) consists of an objective function

$$c_1x_1 + \cdots + c_mx_m$$

and a number of constraints

$$a_{j1}x_1 + \cdots + a_{jm}x_m \ \leq \ b_j, \quad \text{for } j = 1, \dots, k,$$

where $c_1, \dots, c_m \in \mathbb{Z}$, and $a_{j1}, \dots, a_{jm}, b_j \in \mathbb{Z}$, for $j = 1, \dots, k$.
The task is to design a language L_{IP} for integer programs.

a) Specify the alphabet Σ_{IP} you use.

b) Specify how the language L_{IP} is defined.

c) Describe how one can check whether a given word $w \in \Sigma_{\mathrm{IP}}^*$ is in L_{IP}, and, if so, how the integer program can be reconstructed.

End of Exercise 1

Exercise 2: Consider the following problem.

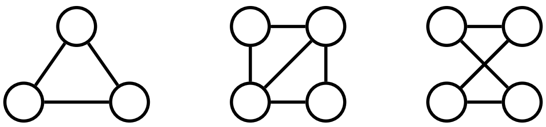
Problem [SubgraphIsomorphism]
Input: Two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.
Output for the decision version: YES if there exists a subgraph $G' = (V', E')$ of G_1 induced by some subset $V' \subseteq V$ such that the graphs G' and G_2 are isomorphic, and NO otherwise.
Output for the optimization version: A subgraph G' as described above if it exists and NO otherwise.

(continued on next page)

Definition of possibly unknown terms:

- Two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *isomorphic* if $|V_1| = |V_2|$ and there is an injective mapping $f\colon V_1 \to V_2$ such that for all $v, w \in V_1$ it holds that $\{v, w\} \in E_1 \iff \{f(v), f(w)\} \in E_2$. Informally, this means that the two graphs are identical after re-naming of the nodes.
- Given a subset $V' \subseteq V_1$, the *subgraph* $G' = (V', E')$ of G_1 *induced by* V' satisfies $E' := \{\{v, w\} \in E_1 \mid v \in V' \wedge w \in V'\}$, i. e., it contains all edges that are incident on nodes in V'.

Example: the graph in the middle has two subgraphs (induced by certain vertex subsets) isomorphic to the graph on the left; the graph on the right has no such induced subgraphs.



What you have to do: Show how to convert an algorithm A_d for the decision version of SUBGRAPHISOMORPHISM into a polynomial-time algorithm A_o for the optimization version of the problem.

a) Describe the algorithm A_o. You may call A_d in your algorithm. Each such call counts as one time step.

b) Argue that your algorithm is correct.

c) Analyze the running time of the algorithm. The running time of A_o has to be polynomial in the input size.

End of Exercise 2

Exercise 3: Consider the following problem:

> **Problem** [BINPACKING]
> **Input:** A sequence $s_1, s_2, \ldots, s_n$ of rational numbers, all being greater than 0 and less than 1, and a natural number B.
> **Interpretation:** We want to pack n objects with sizes $s_1, s_2, \ldots, s_n$ into as few bins as possible. Every bin has a capacity of 1. The objects can be be packed into the bins such that there is no space between them. The objects cannot be divided. The sum of the sizes of the objects in a certain bin cannot exceed 1.
> **Output:** YES if the objects can be packed into at most B bins and NO otherwise.

Prove that BINPACKING is in the class $\mathcal{NP}$.

End of Exercise 3

Exercise 4: Consider the following problem:

> **Problem** [TwoCliques]
> **Input:** An undirected graph $G = (V, E)$ and a positive integer k.
> **Output:** YES if G contains at least two vertex-disjoint cliques of size at least k each. and NO otherwise.
> (Two cliques $V_1, V_2 \subseteq V$ are called vertex-disjoint if $V_1 \cap V_2 = \emptyset$).

Prove that TwoCliques is $\mathcal{NP}$-complete. To do this, you may use that the problem is in $\mathcal{NP}$.
Hint: Your reduction could start from Clique and increase the number of vertices and edges by a factor of about 2.

End of Exercise 4

Exercise 5: Consider the following algorithm.

repeat
  $a \leftarrow rand(1, 10)$;
  $b \leftarrow rand(1, 10)$;
  $c \leftarrow 10 \cdot a + b$;
until c is an even number
$print(c)$;

a) What is the expected running time of the algorithm? It suffices to determine the expected number of iterations of the repeat-until loop.

b) What is the probability that the number 13 is printed? What is the probability that the number 42 is printed?

c) What is the expected value of the number that the algorithm prints?
Hint: The summation formula $\sum_{i=0}^{k} i = \frac{k(k+1)}{2}$ might be useful.

End of Exercise 5

Exercise 6: Consider the following six clauses over the boolean variables $\{x_1, x_2, x_3\}$:

$$
\begin{aligned}
c_1 &= x_1 \vee \overline{x_2} \\
c_2 &= x_3 \\
c_3 &= x_1 \vee x_2 \vee \overline{x_3} \\
c_4 &= \overline{x_1} \vee \overline{x_2} \vee x_3 \\
c_5 &= \overline{x_2} \vee \overline{x_3} \\
c_6 &= \overline{x_1} \vee x_2
\end{aligned}
$$

A truth assignment to x_1, x_2, x_3 can satisfy at most 5 clauses.

a) Construct the relaxed linear program for the set of clauses as described in Section 5.3 of the notes.

b) One solution for a correctly constructed relaxed linear program is
$\hat{z}_1 = 1,\ \hat{z}_2 = 2/3,\ \hat{z}_3 = 1,\ \hat{z}_4 = 1,\ \hat{z}_5 = 1, \hat{z}_6 = 1,$
$\hat{y}_1 = 1/3,\ \hat{y}_2 = 1/3,\ \hat{y}_3 = 2/3$, which gives a target function value of $5\frac{2}{3}$.

Suppose you have access to a random number generator which produces random real numbers in the interval $[0, 1]$ according to uniform distribution.

What you have to do: Apply randomized rounding to find a truth assignment for x_1, x_2, and x_3, where the values given by the random number generator are 0.2234, 0.3422, 0.6943 respectively. Explain how you compute the truth assignment, and show how many clauses it satisfies.

End of Exercise 6

Exercise 7: [Multiple choice]
Answer the multiple choice questions in Figure 1 on the next page. There are 10 questions in total. An incorrect answer is counted negatively, that is, it cancels out a correct answer. However, the total number of points will be at least 0.
The page has to be handed in as part of the solution.

End of Exercise 7

1.)	If a problem is in the class $\mathcal{ZPP}$ then it is not in the class $\mathcal{NP}$.	◯ True	◯ False
2.)	The expression $rand(1,2) \neq rand(3,4)$ is always true.	◯ True	◯ False
3.)	Monte Carlo Algorithms never give a wrong answer.	◯ True	◯ False
4.)	If a randomized algorithm for a decision problem has one-sided error and gives the correct answer with probability exactly 1/2, then it can be converted to one that gives the correct answer with probability exactly 3/4.	◯ True	◯ False
5.)	The primality test described in Section 5.6 of the lecture notes is an $\mathcal{RP}$-algorithm for the problem to decide whether a given integer is not a prime.	◯ True	◯ False
6.)	The expected value printed by the program snippet shown in Figure 2 is 3.5.	◯ True	◯ False
7.)	The probability that $(rand(1,2) = 2 \vee rand(2,3) = 2)$ is true is 3/4.	◯ True	◯ False
8.)	The problem to compute and output a vertex cover of maximal size in an undirected graph is in the class $\mathcal{NP}$.	◯ True	◯ False
9.)	There is a polynomial-time algorithm for KNAPSACK with approximation ratio no worse than 1.01.	◯ True	◯ False
10.)	The randomized search heuristic RLS optimizes every function $f\colon \{0,1\}^n \to \mathbb{R}$ in expected time $O(n^n)$.	◯ True	◯ False

Figure 1: The multiple choice questions.

$i \leftarrow 3$;
    while $(rand(3,4) + rand(1,2) = 5)$ do
    $i \leftarrow i + 1$;
    od
    $print(i)$;

Figure 2: Program snippet for multiple-choice question 6