Computer Science CSC373H
St. George Campus

March 17, 2021
University of Toronto

Homework Assignment #6

Due: March 31, 2021, by 11:00 am

- **You must submit your assignment through the Crowdmark system.** You will receive by email an invitation through which you can submit your work. If you haven't used Crowdmark before, give yourself plenty of time to figure it out!
- You must submit a **separate** PDF document with for **each** question of the assignment.
- To work with one or two partners, you and your partner(s) must form a **group** on Crowdmark (one submission only per group). We allow groups of up to three students, submissions by groups of more than three students will not be graded.
- The PDF file that you submit for each question must be typeset (**not** handwritten) and clearly legible. To this end, we encourage you to learn and use the LaTeX typesetting system, which is designed to produce high-quality documents that contain mathematical notation. You can use other typesetting systems if you prefer, but handwritten documents are not accepted.
- If this assignment is submitted by a group of two or three students, for each assignment question the PDF file that you submit should contain:
  1. The name(s) of the student(s) who *wrote* the solution to this question, and
  2. The name(s) of the student(s) who *read* this solution to verify its clarity and correctness.
- By virtue of submitting this assignment you (and your partners, if you have any) acknowledge that you are aware of the homework collaboration policy for this course, as stated in: `http://www.cs.toronto.edu/~sam/teaching/373/#HomeworkCollaboration`.
- For any question, you may use data structures and algorithms previously described in class, or in prerequisites of this course, without describing them. You may also use any result that we covered in class (in lectures or tutorials) by referring to it.
- Unless we explicitly state otherwise, you should justify your answers. Your paper will be graded based on the correctness and efficiency of your answers, and the clarity, precision, and conciseness of your presentation.
- The total length of your pdf submission should be no more than 4 pages long in a 11pt font.

Question 1. (10 marks) Suppose someone gives you a polynomial-time algorithm *Clique*(G, k) for the Clique ***decision*** problem[1]:

- INPUT: An undirected graph $G = (V, E)$, an integer $k \leq n = |V|$.
- OUTPUT: YES if G has a clique of size k, and NO otherwise.

Give a polynomial-time algorithm *Opt-Clique*(G) that, using the given algorithm *Clique*(G, k), solves this problem:

- INPUT: An undirected graph $G = (V, E)$.
- OUTPUT: A clique $C \subseteq V$ of ***maximum*** size in G.

a. Explain your *Opt-Clique*(G) algorithm in clear and concise English, and then give the pseudo-code. Do not justify its correctness.

b. Assume that the worst-case running time of the given decision algorithm *Clique*(G, k) is some polynomial $p(n)$. What is the running time of your *Opt-Clique*(G) algorithm? Briefly justify your answer.

Question 2. (10 marks) In class we proved that the SAT problem is polynomial-time reducible to the Clique problem (by showing how to transform in polynomial time any instance of the SAT problem into an instance of the Clique problem with the same YES/NO answer).

In this question, you are asked to do the *opposite*: show how to transform (in polynomial time) any instance of the Clique problem into an instance of the SAT problem. To do so, proceed as explained below.

Given any undirected graph $G = (V, E)$ and integer $k \leq n = |V|$, define a Boolean variable x_i^u for every $u \in V$, and every i, $1 \leq i \leq k$; intuitively, $x_i^u = \text{TRUE}$ if and only if "vertex u is the i-th element of some particular k-clique of G". Incrementally construct a CNF formula that it is satisfied if and only if G has a k-Clique, as follows:

a. Write a CNF formula $\mathcal{C}$ that intuitively means:

"For every i, $1 \leq i \leq k$, there is at least one $u \in V$ such that u is the i^{th} element of the clique."

b. Write a CNF formula $\mathcal{D}$ that intuitively means:

"For every i, $1 \leq i \leq k$, no two distinct vertices are both the i^{th} element of the clique."

c. Write a CNF formula $\mathcal{E}$ that intuitively means:

"For every vertex $u \in V$, u is in at most one position in the clique."

d. Write a CNF formula $\mathcal{F}$ that intuitively means:

"If u and v are in the clique, then (u, v) is an edge of G."

HINT: It is equivalent to "if (u, v) is not an edge, then either u is not in the clique or v is not in the clique."

e. Let $\mathcal{B} = \mathcal{C} \wedge \mathcal{D} \wedge \mathcal{E} \wedge \mathcal{F}$; note that this is also a CNF formula.

The size of $\mathcal{B}$ is total number of boolean operators and literals in the formula. What is the size of $\mathcal{B}$ as a function of n? Use the $O()$ notation and briefly justify your answer.

f. Prove that $\mathcal{B}$ is satisfiable if and only if $G = (V, E)$ has a clique of size k.

[1] If this ever happens, please let me know.

Question 3. (12 marks)

a. Suppose your community has several special interest groups, e.g., lawyers, avid motorcyclists, nature lovers, teachers, etc...(note that a person may belong to multiple groups) and you want to form the smallest committee that includes at least one person from each group.

This gives rise to the following ***Inclusive Committee*** problem:

- **Given**: A set S, a collection of subsets $S_1, S_2, \ldots, S_m \subseteq S$, and an integer k.

  Intuitively, S is the set of persons in your community, and each S_i is the subset of persons belonging to special interest group i.

- **Question**: Is there a subset $S' \subseteq S$ of size k or less, such that S' contains at least one element from each of $S_1, S_2, \ldots, S_m$?

  Intuitively, is there a committee S' of size k that includes a person from each special interest group?

Prove that this problem is NP-Complete.

HINT: There is an ***easy*** reduction using one of the problems that we saw in class.

b. Brunella di Montalcino lives in a city where all the streets are one-way (even for walkers), and she likes to walk. Every day she walks a tour that does not repeat any street, or any street corner: she does not want the tour to be boring. Today she wants her tour to include a specific set of streets because she has errands to do there.

This gives rise to the following ***Goal-Oriented Walker*** problem:

- **Given**: A directed graph $G = (V, E)$, and a subset of edges $E^* \subseteq E$.

  Intuitively, edges are the city streets, nodes are the street corners (or end-points of no-exit streets), and E^* are the streets that Brunella wants to include in her tour.

- **Question**: Does G have a simple directed cycle (i.e., a cycle that does not repeat vertices) that contains ***at least*** every edge in E^*?

Prove that this problem is NP-Complete.

HINT: Use the *Directed Hamiltonian Cycle* problem (which is NP-Complete, see page 1049 of CLRS).