COMP 360 - Winter 2021 - Assignment 4

Due: 11:59pm April 16th.

General rules: In solving these questions you may consult books or other available notes, but you need to provide citations in that case. You can discuss high level ideas with each other, but each student must find and write his/her own solution. Copying solutions from any source, completely or partially, allowing others to copy your work, will not be tolerated, and will be reported to the disciplinary office. You should upload the pdf file (either typed, or a clear and readable scan) of your solution to MyCourses.

1. (*5 points*) Suppose you are given a set of positive integers $A = \{a_1, a_2, \ldots, a_n\}$ and a positive integer B. A subset $S \subseteq A$ is called feasible if the sum of the numbers in S does not exceed B:

$$\sum_{a_i \in S} a_i \leq B.$$

The sum of the numbers in S will be called the total sum of S. You would like to select a feasible subset S of A whose total sum is as large as possible.
Give a polynomial-time approximation algorithm for this problem with the following guarantee:
It returns a feasible set $S \subseteq A$ whose total sum is at least half as large as the maximum total sum of any feasible set $S' \subseteq A$. Your algorithm should have a running time of at most $O(n \log n)$.

(**Note:** you need to provide an explanation/proof why your algorithm works.)

2. (*5 points*)

- Show that if an NP-hard problem lies in coNP, then NP $\subseteq$ coNP.
- Show that if $X \leq_P Y$ and $Y \in coNP$ then $X \in coNP$.

3. (*5 points*) A new instructor claimed that the following problem is in NP-hard but it is not NP-complete. What do you think about this claim?

- Input: An undirected graph G, two nodes s, t and a positive integer k.
- Question: Does the shortest path between s and t has length exactly k?

4. (*5 points*) Show that if a problem is in PSPACE-complete, then it is in NP-hard.

5. (*10 points*) Consider the triangle elimination problem. We are given an undirected graph $G = (V, E)$, and want to find the smallest possible set of vertices $U \subseteq V$ such that deleting these vertices removes all the triangles (i.e. cycles of length 3) from the graph. For each one of the following algorithms, either show that it is a 3-factor approximation algorithm, or give an example to show that it is not.

Algorithm I:

- While there is still a triangle C left in G:
- Delete all the three vertices of C from G.
- EndWhile
- Output the set of the deleted vertices.

Algorithm II:

- While there is still a triangle C left in G:
- Delete one arbitrarily chosen vertex of C from G.
- EndWhile
- Output the set of the deleted vertices.