ECS 122B: Conceptual Homework #3

Instructor: Aaron Kaloti

Summer Session #2 2020

1 Changelog

You should always refer to the latest version of this document.

- v.1: Initial version.

2 Grading

- **Due date**: the night of Friday, 08/28. (Due to the "grace period", Gradescope will say 12:30 AM on Saturday, 08/29.)
- A subset of the problems will be graded for correctness. The rest will be graded on completion.

3 Submission Requirements

- Your submission must be created with LaTeX. **Handwritten/scanned solutions, or solutions not typed with LaTeX, will earn no credit.**
  - You should avoid doing this assignment at the last minute, because it will take a bit of time to get used to LaTeX and to type your answers up into a LaTeX document.
- Your submission must consist of one file, the PDF generated by your LaTeX file (`hw3_answers.pdf`). Since Gradescope does not let you submit files that are neither PDFs nor images, I will not have you submit the `.tex` file. However, if it is clear that your submission was not made with LaTeX, we may email you to ask that you send us your `.tex` file, and if you cannot (i.e. if it turns out that your PDF was in fact not created with LaTeX), you will get a zero on this assignment.
- You may be penalized if, when submitting on Gradescope, you mark the wrong page for a given homework problem, because dealing with this slows down grading. At the beginning of the 08/06 lecture, I talk about how to mark the pages of your submission on Gradescope, starting at around 3:20 in the video.
- When using LaTeX, you should make use of the math notation/mode where sensible. (The math mode refers to when you put mathematical equations, etc. between dollar signs so that LaTex makes them look nice.) Repeated failures to do this, to the point that the purpose of using LaTeX is defeated or the readability of your answers is impeded, may result in a penalty on this assignment.

4 Regarding Collaboration

- You may not copy answers from any sources, including online sources such as Chegg, StackOverflow, or any solutions manual of any textbook.
- You may partner with ***at most one*** other student on this assignment. In other words, you can work in *pairs*. You do not have to partner with anyone. (In fact, I think it is better that you do not.) If you partner with someone, it must be a committed partnership; that is, you two will have the same submission, and you must mark on Gradescope that you have partnered for this assignment by following the directions here.
- If students that were not in the same pair seem to have excessively similar answers, they will be reported to the OSSJA for suspicion of academic misconduct. Do not copy answers from (or share answers with) any student who you are not partnered with.

5 Identification

Enter the members of your pair. (You can partner with at most one other student.) If you are not partnered with anyone, then leave the second box empty. You can remove the use of `\vspace` in the `.tex` file.

Pair member #1:

Julio Beas

Pair member #2:

Han Nguyen

6 Problems

Place your answer into the answer boxes; you can remove the use of `\vspace` in the answer boxes in the `.tex` file. If you are using your own LaTeX template, you may find the `setcounter` directive useful; it lets you set the number of the next section, and you can read more here.

6.1 k-independent set $\leq_P$ k-clique

As stated in p.1086 of *Introduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein (Third Edition), "a ***clique*** in an undirected graph $G = (V, E)$ is a subset $V\prime \subseteq V$ of vertices, each pair of which is connected by an edge in E. In other words, a clique is a complete subgraph of G. The ***size*** of a clique is the number of vertices it contains." In the k-Clique Problem, we ask: Given an undirected graph $G = (V, E)$ and an integer k, does there exist a clique of at least size k?

Given that the k-Independent Set Problem is $\mathcal{NP}$-hard, show that the k-Clique Problem is $\mathcal{NP}$-hard. This will involve a polynomial-time reduction, and you should address this with some rigor, i.e. you must show that the reduction takes polynomial time, and you must prove the things we usually prove in reductions.

First we will convert an instance of k-independent set as follows. An instance of k-independent set is a graph G an and integer k. Construct G^c, the complement of G, and pass G^c, k to k-clique. Next we have to show that "yes" instance of k-independent set maps to a "yes" instance of k-clique and vice versa. Assume G is a "yes" instance of k-independent set, so there exists a k-independent set (I) of size k in G. Thus, for any $u, v \,\epsilon$ I, (u,v) $\not\epsilon$ E. Thus, (u,v) $\not\epsilon\, E^c$. Thus, the vertices in I form and independent set in G^c. So, G^c is a "yes" instance of k-clique. Next assume G^c is a "yes" instance of k-clique set, so there exists a k-clique set (C) of size k in G^c. Thus, for any $u, v \,\epsilon$ C, (u,v) $\not\epsilon\, E^c$. Thus, (u,v) ϵ E. Thus, the vertices in C form a k-independent set in G. So, G is a "yes" instance of k-independent set.

6.2 set partition $\leq_P$ subset sum

The Subset Sum Problem is as follows: Given a set S of positive integers and a *number* (not necessarily an integer) target $t > 0$, does there exist a subset $S\prime \subseteq S$ whose elements sum to t? For example, if $S = \{3, 8, 15, 29\}$ and $t = 23$, then the answer is "yes", as shown by $S\prime = \{8, 15\}$.

The Set Partition Problem is as follows: Given a *set* (not multiset) S of numbers, can the numbers be partitioned into two sets A and $\bar{A} = S - A$ such that $\sum\limits_{x \in A} x = \sum\limits_{x \in \bar{A}} x$. For example, if $S = \{2, 3, 5, 6\}$, then the answer is "yes", as shown by $A = \{2, 6\}$ and $\bar{A} = S - A = \{3, 5\}$.

Given that the Set Partition Problem is $\mathcal{NP}$-hard, show that the Subset Sum Problem is $\mathcal{NP}$-hard. This will involve a polynomial-time reduction, and you should address this with some rigor, i.e. you must show that the reduction takes polynomial time, and you must prove the things we usually prove in reductions.

First, we will show that Subset Sum Problem is at least as hard as Set Partition Problem. We will check the reducibility of Set Partition from Subset Sum takes polynomial time. From the above, we will input $S' = S \cup \{s - 2t\}$ to the Subset Sum Problem. This clearly takes polynomial time.
Finally, we will prove that $(S') \in$ Set Partition implies $(S, t) \in$ Subset Sum. Suppose there exists a partition of S' into two sets whose sum of each is $s - t$. Hence, since the sum of each items in S' is $2s - 2t$, we say that one of these two sets must have the number $s - 2t$. Hence, as $S = S' - \{s - 2t\}$ from the above, we conclude that once removing $s - 2t$ from S', we will yield a set of numbers whose sum is t, which is S. This means once $(S, t) \in$ Subset Sum, the case that $(S') \in$ Set Partition still holds.
Therefore, from the reduction above, we say that Set Partition is reducible by Subset Sum. Hence, Subset Sum is $\mathcal{NP}$-hard.