COP 6405: Analysis of Algorithms FINAL REVIEW - PART 3; Fall 2019

---

1. **Short Questions**
   (a) What are the main consequences of a polynomial-time reduction from problem Π_1 to Π_2?
   (b) Why does Cook's Theorem show that SAT is the hardest problem in $\mathcal{NP}$?
   (c) Write down the decision version of the *minimum spanning tree* problem.
   (d) Differentiate between *NP-Complete* and *NP-hard*. Note that the optimization version of every NP-complete decision problem is NP-hard.
   (e) Correct the following sentence as needed: "To prove a polynomial-time reduction from decision problem Π_1 to decision problem Π_2, one needs to design a polynomial-time algorithm that takes an arbitrary instance of problem Π_1 (call it π_1) and creates an instance of problem Π_2 (call it π_2) in such a way that π_1 is a YES-instance of Π_1 if π_2 is a YES-instance of Π_2."
   (f) The well-known NP-complete decision problem CLIQUE is formally described here:
   INSTANCE: Graph $G = (V, E)$, positive integer $k \leq |V|$.
   QUESTION: Does G contain a clique of size k or more, i.e., a subset $V' \subseteq V$ with $|V'| \geq k$ such that every pair of vertices in $V'$is joined by an edge in E?
   Now write down the NP-complete problem TSP in the same format. It is necessary to be mathematically precise in this formulation.
   (g) Write down the VERTEX COVER problem in the above format.
   (h) What does one have to do to prove that a problem is in $\mathcal{NP}$?
   (i) Your friend claims that he has proved that problem Π is NP-hard, because he has a polynomial-time reduction from Π to SAT. Please explain to him why his proof is faulty.
   (j) Name as many open problems as you can related to the classes $\mathcal{P}$, $\mathcal{NP}$, co-$\mathcal{NP}$, and $\mathcal{NP}$-complete.
2. Read and understand Section 34.5.2 that shows that the VERTEX COVER problem is NP-Complete using a polynomial-time reduction from the CLIQUE problem.
3. Read and understand Section 34.5.4 that shows that the TSP problem is NP-Complete using a polynomial-time reduction from the HAMILTONIAN-CYCLE problem.
4. Read and understand the problem description for the SUBSET-SUM problem (start of Section 34.5.4). Now show that SUBSET-SUM is in $\mathcal{NP}$ (not $\mathcal{NP}$-complete).