CS 6363.003 Homework 5

Due Friday, May 5 on eLearning

Please solve the following 4 problems, some of which have multiple parts.

1. A given flow network \( G = (V, E) \) with source \( s \), sink \( t \), and capacity function \( c : E \to \mathbb{R}_{\geq 0} \) may have more than one minimum \((s, t)\)-cut.

   (a) Let \((S, T)\) and \((S', T')\) be two minimum \((s, t)\)-cuts in \( G \). Prove that \((S \cap S', T \cup T')\) and \((S \cup S', T \cap T')\) are also minimum \((s, t)\)-cuts in \( G \). [Hint: Let \( f^* \) be a maximum \((s, t)\)-flow in \( G \). What have we learned about \( f^* \) and the edges crossing any minimum \((s, t)\)-cut?]

   (b) Describe and analyze an efficient algorithm to determine whether \( G \) contains a unique minimum \((s, t)\)-cut. [Hint: Use the claim from part (a). Modify the process for finding a minimum \((s, t)\)-cut given a maximum \((s, t)\)-flow.]

2. The new Department of Computing Stuff at UTD has a flexible curriculum with a complex set of graduation requirements. The department offers \( n \) different courses, and there are \( m \) different requirements. Each requirement specifies a subset of the \( n \) courses and the number of courses that must be taken from that subset. The subsets for different requirements may overlap, but each course can be used to satisfy at most one requirement.

   For example, suppose there are \( n = 5 \) courses \( A, B, C, D, \) and \( E \) and \( m = 2 \) graduation requirements:

   - You must take at least 2 courses from the subset \( \{A, B, C\} \).
   - You must take at least 2 courses from the subset \( \{C, D, E\} \).

   Then a student who has only taken courses \( B, C \), and \( D \) cannot graduate, but a student who has taken either \( A, B, C \), and \( D \) or \( B, C, D, \) and \( E \) can graduate.

   Describe and analyze an algorithm to determine whether a given student can graduate. The input to your algorithm is the list of \( m \) requirements (each specifying a subset of the \( n \) courses and the number of courses that must be taken from that subset) and the list of courses the student has taken.

3. A boolean formula is in **disjunctive normal form** (DNF) if it consists of a **disjunction** (Or) of several **terms**, each of which is the conjunction (And) of one or more literals. For example, the formula

   \[
   (\bar{x} \land y \land \bar{z}) \lor (y \land z) \lor (x \land \bar{y} \land \bar{z})
   \]

   is in disjunctive normal form. DNF-SAT asks, given a boolean formula in disjunctive normal form, whether that formula is satisfiable.
(a) Describe a polynomial-time algorithm to solve DNF-SAT.

(b) What is the error in the following argument that $P = NP$?

Suppose we are given a boolean formula in conjunctive normal form with exactly three literals per clause, and we want to know if it is satisfiable. We can use the distributive law to construct an equivalent formula in disjunctive normal form. For example,

$$(x \lor y \lor \bar{z}) \land (\bar{x} \lor \bar{y} \lor z) \iff (x \land \bar{y}) \lor (x \land z) \lor (y \land \bar{z}) \lor (y \land \bar{x}) \lor (z \land \bar{x}) \lor (z \land \bar{y})$$

Now we can use the algorithm from part (a) to determine, in polynomial time, whether the resulting DNF formula is satisfiable. We have just solved 3Sat in polynomial time. Since 3Sat is NP-hard, we must conclude that $P = NP$!

4. Both of the following games involves an $n \times m$ grid of squares, where each square is either empty or occupied by a stone. In a single move, you can remove all the stones in an arbitrary column.

(a) Prove that it is NP-hard to find the smallest subset of columns that can be cleared so that at most one stone remains in each row of the grid.

(b) Prove that it is NP-hard to find the largest subset of columns that can be cleared so that at least one stone remains in each row of the grid.