CSC364 Summer 2004 — Homework 4

The following questions are assigned each week. None of the questions are for hand in, but you are encouraged to try them.

Week 12

1. Let 2-CNF-SAT be the set of satisfiable boolean formulas in CNF with exactly 2 literals per clause. Show that 2-CNF-SAT $\in$ **P**. Make your algorithm as efficient as possible.
   Hint: Observe that $x \vee y$ is equivalent to $\neg x \rightarrow y$. Reduce 2-CNF-SAT to a problem on directed graph that is efficiently solvable.

2. We define the following three related problems.

   5-CLIQUE
   Instance: $\langle G \rangle$ where G is an undirected graph.
   Acceptance Condition: Accept iff G contains a 5-clique.

   CLIQUE
   Instance: $\langle G, k \rangle$, G is an undirected graph, k an integer in binary.
   Acceptance Condition: Accept iff G has a k-clique.

   MAX-CLIQUE
   Instance: $\langle G \rangle$ where G is an undirected graph.
   Output: Return the size of the largest clique of G.

   (a) Give a polynomial time algorithm that solves 5-CLIQUE.

   (b) Prove MAX-CLIQUE $\xrightarrow{p}$ CLIQUE.
   This proves we can solve MAX-CLIQUE with a polynomial number of calls to CLIQUE.

   (c) Prove CLIQUE $\xrightarrow{p}$ MAX-CLIQUE.
   This shows that MAX-CLIQUE is likely not in **FP** since CLIQUE is **NP**-complete.

   (d) Consider the following algorithm for MAX-CLIQUE:

   ```
   MAX-CLIQUE(G):
   for i ← 1 to n do
       if not i-CLIQUE(G) then
           return i − 1
   return n
   ```

   We proved in part (a) that i-CLIQUE is in P, and we only make a polynomial number of calls to i-CLIQUE. Why does this not prove MAX-CLIQUE is in **FP**?

Week 13

3. Linear programming when we restrict the variables to be zero or one is called **0-1 integer programming**. Specifically, the 0-1 integer programming problem asks, when given an integer $m \times n$ matrix A and an integer m-vector b, whether there exists an n-vector whose elements are selected from $\{0, 1\}$ such that $Ax \leq b$.
   Prove that 0-1 integer programming is **NP**-complete.
   Hint: Try a reduction from 3-SAT.

4. Prove 2-COLOR $\in$ **P**.
   2-COLOR is given a graph G, can we color the vertices of G with 2 colors such that adjacent vertices get different colors?