CPSC 320: Tutorial 7

1. (from Problem 34-1 CLRS) An *independent set* of a graph $G = (V, E)$ is a subset $S \subset V$ of vertices such that no two vertices in S are joined by an edge. We want to find the maximum size independent set in G, but all our attempts at an efficient algorithm have failed.

   - Specify a related decision problem for this problem and prove that it is NP-complete. (Hint: Reduce from CLIQUE.)
   - How would you use an efficient algorithm for your decision problem to solve the original problem efficiently? In other words, can you find a maximum size independent set efficiently by using a black box that efficiently solves your decision problem?
   - What does this say about the likelihood of finding an efficient algorithm for the original problem?

2. A *k-clique* in a graph is a set of k vertices which induce a k-complete graph. The question of whether a k-clique exists in a graph is, in general NP-complete.

   Prove by reduction that CLIQUE is NP-complete.

3. (from Problem 34-3 CLRS) *Graph colouring*

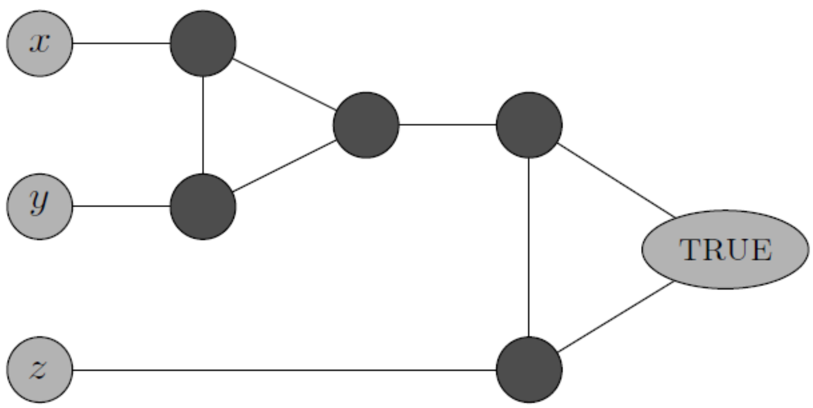
   A *k-colouring* of an undirected graph $G = (V, E)$ is a function $c : V \longrightarrow \{1, 2, \ldots k\}$ such that $c(u) \neq c(v)$ for every edge $(u, v) \in E$. In other words, the numbers $1, 2, ..., k$ represent the k colours, and adjacent vertices must have different colours. The *graph-colouring problem* is to determine the minimum number of colours needed to colour a given graph.

   - Cast the graph-colouring problem as a decision problem. Show that your decision problem is solvable in polynomial time if and only if the graph-colouring problem is solvable in polynomial time.
   - Let the language 3-colour be the set of graphs that can be 3-coloured. Show that if 3-colour is NP-complete, then your decision problem is NP-complete.

To prove that 3-colour is NP-complete, we use a reduction from 3-SAT. Given a formula ϕ of m clauses on n variables $x_1, x_2, \ldots, x_n$, we construct a graph $G = (V, E)$ as follows. The set V consists of a vertex for each variable, a vertex for the negation of each variable, 5 vertices for each clause, and 3 special vertices: TRUE, FALSE, and RED. The edges of the graph are of two types: "literal" edges that are independent of the clauses and "clause" edges that depend on the clauses. The literal edges form a triangle on the special vertices and also form a triangle on $x_i, \overline{x_i}$, and RED for $i = 1, 2, \ldots, n$.

- In any 3-colouring c of a graph containing the literal edges, exactly one of a variable and its negation is coloured $c(\text{TRUE})$ and the other is coloured $c(\text{FALSE})$. Why?
- Any truth assignment (whether it satisfies ϕ or not) corresponds to a 3-colouring of the graph formed from just the literal edges. Why?

The widget shown below is used to enforce the condition corresponding to a clause $(x \vee y \vee z)$. Each clause requires a unique copy of the 5 vertices that are heavily shaded in the figure; they connect as shown to the literals of the clause and the special vertex TRUE.



- Argue that if each of x, y, and z is coloured $c(\text{TRUE})$ or $c(\text{FALSE})$, then the widget is 3-colourable if and only if at least one of x, y, or z is coloured $c(\text{TRUE})$.

- Complete the proof that 3-colour is NP-complete.