## Practice problems (don't turn in):

- 1. [DPV] Problem 8.1 (TSP optimization versus search)
- 2. [DPV] Problem 8.3 (Stingy SAT)
- 3. [DPV] Problem 8.4 (a),(b),(c) (Clique-3)
- 4. [DPV] Problem 8.10 part (a) (Subgraph isomorphism)
- 5. This problem is designed to make think about the following idea: sometimes a combination (or modification) of a hard problem is not necessarily more difficult. Problem 8.14 explore this idea further.

Design an efficient algorithm to solve the following problem:

Input: A directed graph G = (V, E). Output: A cycle if G has one, or a path that visits all vertices, if there is one.

Note that your algorithm will output NO only if there are no cycles in G and there is no path visiting all vertices.

## Problem 1 (Almost-SAT)

The Almost-SAT problem takes as input a boolean formula on n literals, in conjunctive normal form with m clauses. The output is an assignment of the literals such that **exactly** m - 1 clauses evaluates to TRUE, if such assignment exists, and outputs NO otherwise. Show that Almost-SAT is NP-complete.

Problem 2 [DPV] Problem 8.14 (Clique+IS)