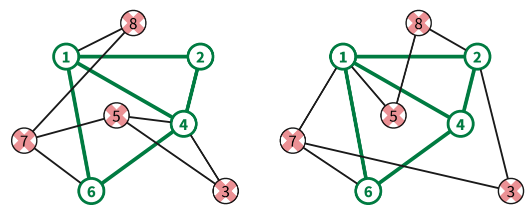
CS/ECE 374 A ✧ Fall 2021

Homework 10

Due Tuesday, November 16, 2021 at 8pm

---

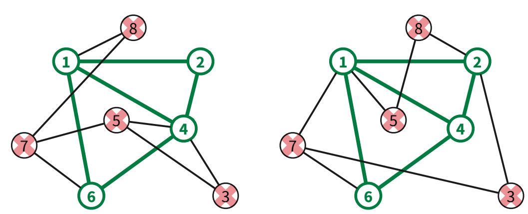
1. Suppose we are given two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ with the same set of vertices $V = \{1, 2, \ldots, n\}$. You are given the following problem: find the smallest subset $S \subseteq V$ of vertices whose deletion leaves identical subgraphs $G_1 \setminus S = G_2 \setminus S$. For example, given the graphs below, the smallest subset has size 4.



Provide a polynomial-time reduction for this problem from *any one of the following three* problems:

- MaxIndependentSet: MaxIndependentSet(G, m) returns 1 if the size of the largest independent set in graph G is m, otherwise returns 0.
- MaxClique: MaxClique(G, m) returns 1 if the size of the largest clique in G is m, otherwise returns 0.
- MinVertexCover: MinVertexCover(G, m) returns 1 if the size of the smallest vertex cover in G is m, otherwise returns 0.

Hint: There exists a reduction to all three problems; you may pick whichever one is most convenient for you.



2. This problem asks you to develop polynomial-time algorithms for two (apparently) minor variants of 3SAT.

   (a) The input to **2SAT** is a boolean formula Φ in conjunctive normal form, with exactly **two** literals per clause, and the 2SAT problem asks whether there is an assignment to the variables of Φ such that every clause contains at least one TRUE literal.

   Describe a polynomial-time algorithm for 2SAT. *[Hint: This problem is strongly connected to topics covered earlier in the semester.]*

   (b) The input to **MAJORITY3SAT** is a boolean formula Φ in conjunctive normal form, with exactly three literals per clause. MAJORITY3SAT asks whether there is an assignment to the variables of Φ such that every clause contains *at least two* TRUE literals.

   Describe and analyze a polynomial-time reduction from MAJORITY3SAT to 2SAT. Don't forget to prove that your reduction is correct.

   (c) Combining parts (a) and (b) gives us an algorithm for MAJORITY3SAT. What is the running time of this algorithm?

Solved Problem

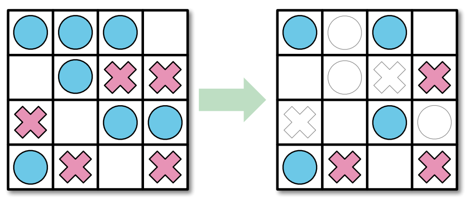
3. Consider the following solitaire game. The puzzle consists of an $n \times m$ grid of squares, where each square may be empty, occupied by a red stone, or occupied by a blue stone. The goal of the puzzle is to remove some of the given stones so that the remaining stones satisfy two conditions:

   (1) Every row contains at least one stone.

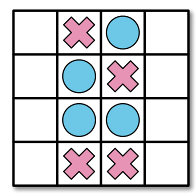
   (2) No column contains stones of both colors.

   For some initial configurations of stones, reaching this goal is impossible; see the example below.

   Prove that it is NP-hard to determine, given an initial configuration of red and blue stones, whether this puzzle can be solved.



A solvable puzzle and one of its many solutions.



An unsolvable puzzle.

Solution: We show that this puzzle is NP-hard by describing a reduction from 3SAT.

Let Φ be a 3CNF boolean formula with m variables and n clauses. We transform this formula into a puzzle configuration in polynomial time as follows. The size of the board is $n \times m$. The stones are placed as follows, for all indices i and j:

- If the variable x_j appears in the ith clause of Φ, we place a blue stone at (i, j).
- If the negated variable $\overline{x_j}$ appears in the ith clause of Φ, we place a red stone at (i, j).
- Otherwise, we leave cell (i, j) blank.

We claim that this puzzle has a solution if and only if Φ is satisfiable. This claim immediately implies that solving the puzzle is NP-hard. We prove our claim as follows:

$\Longrightarrow$ First, suppose Φ is satisfiable; consider an arbitrary satisfying assignment. For each index j, remove stones from column j according to the value assigned to x_j:

- If $x_j = \text{TRUE}$, remove all red stones from column j.
- If $x_j = \text{FALSE}$, remove all blue stones from column j.

In other words, remove precisely the stones that correspond to FALSE literals. Because every variable appears in at least one clause, each column now contains stones of only one color (if any). On the other hand, each clause of Φ must contain at least one TRUE literal, and thus each row still contains at least one stone. We conclude that the puzzle is satisfiable.

$\Longleftarrow$ On the other hand, suppose the puzzle is solvable; consider an arbitrary solution. For each index j, assign a value to x_j depending on the colors of stones left in column j:

- If column j contains blue stones, set $x_j = \text{True}$.
- If column j contains red stones, set $x_j = \text{False}$.
- If column j is empty, set x_j arbitrarily.

In other words, assign values to the variables so that the literals corresponding to the remaining stones are all True. Each row still has at least one stone, so each clause of Φ contains at least one True literal, so this assignment makes $\Phi = \text{True}$. We conclude that Φ is satisfiable.

This reduction clearly requires only polynomial time. ■

Rubric (Standard polynomial-time reduction rubric): 10 points =

+ 3 points for the reduction itself
  - For an NP-hardness proof, the reduction must be from a known NP-hard problem. You can use any of the NP-hard problems listed in the lecture notes (except the one you are trying to prove NP-hard, of course). **See the list on the next page.**
+ 3 points for the "if" proof of correctness
+ 3 points for the "only if" proof of correctness
+ 1 point for writing "polynomial time"

- An incorrect polynomial-time reduction that still satisfies half of the correctness proof is worth at most 4/10.
- A reduction in the wrong direction is worth 0/10.