ECE 374 B - Fall 2025

Homework 8

The instructions below are the typical instructions I give for homeworks in prior semesters. In the Fall 2025 semester, I decided to do away with homeworks and simply calculate the course grade based on quiz and exam performance. However, I am still releasing these homeworks so students may have some extra study materials. We **are not** collecting/grading homework assignments in the Fall 2025 semester.

Word of advice: before working on these homeworks ask yourself: "Can I do all the lab problems right now (without looking at the solutions)?" If the answer is no, that is where you need to spend your time. If the answer is yes, then feel free to proceed. But again, don't be dumb and look at the solutions first. Lab/homework problems are opportunities to learn and you learn by struggling through them. Looking at a solution and telling yourself "Yeah, I get it" is the best way to do poorly in this course.

- **Every homework problem must be done *individually*.** Each problem needs to be submitted to Gradescope before 6AM of the due data which can be found on the course website: https://ecealgo.com/homeworks.html.
- For nearly every problem, **we have covered all the requisite knowledge required to complete a homework assignment prior to the "assigned" date.** This means that there is no reason *not* to begin a homework assignment as soon as it is assigned. Starting a problem the night before it is due a recipe for failure.

Policies to keep in mind

- **You may use any source at your disposal**—paper, electronic, or human—but you ***must*** cite ***every*** source that you use, and you ***must*** write everything yourself in your own words. See the academic integrity policies on the course web site for more details.
- **Being able to clearly and concisely explain your solution is a part of how you are graded in this course and elsewhere.** Before submitting a solution ask yourself, if you were reading the solution without having seen it before, would you be able to understand it within two minutes? If not, you need to edit. Images and flow-charts are very useful for concisely explain difficult concepts.

See the course web site (https://ecealgo.com/) for more information.

If you have any questions about these policies,
please don't hesitate to ask in class, in office hours, or on Piazza.

1. Professor Amongus is researcher that keeps trying to prove $P = NP$.

   (a) First he showed that a decision problem L is polynomial-time reducible to an NP-complete problem M. Moreover, after 80 pages of dense mathematics, he has also just proven that L can be solved in polynomial time. Has he just proven that P = NP ? Why, or why not?

   (b) Next, he designed an algorithm that can take any graph G with n vertices and determine in O(n^k) time whether G contains a clique of size k. Does Professor Amongus deserve the Turing Award for having just shown that P = NP? Why or why not?

2. **Cyclic paths of hell.** A Hamiltonian cycle in a graph is a cycle that visits every vertex exactly once. A Hamiltonian path in a graph is a path that visits every vertex exactly once, but it need not be a cycle (the last vertex in the path may not be adjacent to the first vertex in the path.)

   Consider the following three problems:

   - *Directed Hamiltonian Cycle* problem: checks whether a Hamiltonian cycle exists in a *directed* graph,
   - *Undirected Hamiltonian Cycle* problem: checks whether a Hamiltonian cycle exists in an *undirected* graph.
   - *Undirected Hamiltonian Path* problem: checks whether a Hamiltonian path exists in an *undirected* graph.

a Give a polynomial time reduction from the *directed* Hamiltonian cycle problem to the *undirected* Hamiltonian cycle problem.

b Give a polynomial time reduction from the *undirected* Hamiltonian Cycle to *directed* Hamiltonian cycle.

c Give a polynomial-time reduction from undirected Hamiltonian *Path* to undirected Hamiltonian *Cycle*.

3. We have the following two problems:

   - LONGEST-PATH-LENGTH - you are given a undirected graph and two vertices s, t and returns the number of edges in the longest simple path between the two vertices.
   - LONGEST-PATH - you are given a graph $G = (V, E)$, two vertices $u, v \in V$, an integer $k \geq 0$, and the problem outputs whether or not there is a simple path from u to v in G containing atleast k edges.

   Show that the optimization problem, LONGEST-PATH-LENGTH can be solved in polynomial time if and only if LONGEST-PATH $\in P$

4. For each of the following problems, pick true or false and explain why.

   (a) **True/False:** If L is an NP-complete language and $L \in P$, then $P = NP$.

   (b) **True/False:** There exists a polynomial-time reduction from every problem in NP to every problem in P.

   (c) **True/False:** If a problem is both NP-hard and co-NP-hard, then it must be in NP.

   (d) **True/False:** If there is a polynomial-time reduction from problem A to problem B and B is in NP, then A must also be in NP.

   (e) **True/False:** If a problem is solvable in polynomial space, then it is also solvable in polynomial time.