CECS 329, Homework Assignment 2, Spring 2026, Dr. Ebert

Directions: Please review the `Homework` section on page 6 of the syllabus including a list of all rules and guidelines for writing and submitting solutions.

Due Date: Tuesday February 24th as a PDF-file upload to the HW2 Canvas dropbox.

Problems

1. Recall that a **Hamilton Cycle (HC)** for a graph $G = (V, E)$ is a path that starts at some vertex $v \in V$, visits every other vertex of G exactly once, and then returns back to v to complete the cycle. As a traffic engineer, Frank sometimes has to compute Hamilton Cycles within transportation networks. Because of this, he has written a program that can decide if a given graph $G = (V, E)$ has an HC. His colleague Jennifer is analyzing a transportation network, modeled as a graph H, and she is interested in deciding if H has a Hamilton Path (HP) which starts at one particular location a, visits every vertex, and ends at a different location b. Frank explained to Jennifer that she could use his HC program to decide if H has an HP. "Simply connect your a and b vertices with an edge. That way, your updated graph having an HC will be equivalent to your original graph H having an HP". Give an example that shows that Frank's statement is not always true. Conclude that Frank's method is not a valid way to map reduce HP to HC. (15 pts)

2. This problem is inspired by my supervisor at Arcadia Design Systems who told me "linear and log-linear [algorithms]: good. Anything else, be very careful how you use it". Suppose that a computer's cpu is capable of executing a single instruction in 5×10^{-10} seconds. The plan is to run a time-intensive program on the computer for 30 days. For each of the following scenarios determine the largest problem size n that can be solved during this time. Show all steps for full credit. Please round down each answer to the nearest integer. All answers should be accurate to the nearest integer.

   (a) The program solves instances of `Matrix Addition` and requires about $100n^2$ instructions to add two $n \times n$ matrices. (7 pts)

   (b) The program solves instances of `Matrix Multiplication` and requires about $180n^3$ instructions to multiply two $n \times n$ matrices. (7 pts)

   (c) The program solves instances of the `3SAT` logic problem and implements the best known `3SAT` algorithm which requires about $50\left(\frac{4}{3}\right)^n$ instructions to solve a `3SAT` instance that depends on n variables. (7 pts)