CS182 – Foundations of Computer Science

PSO sessions 1 and 2, week of February 24, 2020

PSO 1

Problem 1. Find a 2×2 matrix A that is not identically zero (e.g., not all its entries are equal to zero) such that $A^2 = 0$.

Problem 2. Let $A = \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix}$. Find A^n for any $n \geq 1$.

Problem 3. Suppose $A = \begin{pmatrix} 3 & 5 \\ 2 & 4 \end{pmatrix}$ and $C = \begin{pmatrix} 2 & 0 \\ 1 & 6 \end{pmatrix}$. Find a matrix B such that $AB = C$ or prove that no such matrix exists.

Problem 4. Suppose $B = \begin{pmatrix} 3 & 5 \\ 2 & 4 \end{pmatrix}$ and $C = \begin{pmatrix} 2 & 0 \\ 1 & 6 \end{pmatrix}$. Find a matrix A such that $AB = C$ or prove that no such matrix exists.

Problem 5. Is the following statement true or false: if $AB = AC$ then $B = C$. Start by considering the case where A, B, and C are 1×1 matrices and proceed.

Task 1. Use any remaining time as office hours.

PSO 2

Problem 1. Express a brute-force algorithm that finds the largest product of two numbers in a list $a_1, a_2, \ldots, a_n$ with $n \geq 2$ that is less than a threshold N.

Problem 2. You have supplies of boards that are one foot, five feet, seven feet, and twelve feet long. You need to lay pieces end-to-end to make a molding 15 feet long and wish to do this using the fewest number of pieces possible. Explain why the greedy algorithm of taking boards of the longest length at each stage (so long as the total length of the boards selected does not exceed 15 feet) does not give the fewest number of boards possible.

Problem 3. Prove or disprove that the greedy algorithm for making change always uses the fewest coins possible when the denominations available are pennies (1-cent coins), nickels (5-cent coins), and quarters (25-cent coins).

Task 1. Use any remaining time as office hours.