CS3414. Homework problem set VI.

1 Problem I. 20 points

A good CS3414 student needs to find the inverse A^{-1} of an arbitrary non-singular $n \times n$ matrix A using the following algorithm: solve $\mathbf{A}\ \vec{x}_k = \vec{I_k}$ for each $k = 1, 2, ...n$. Here x_k is the k-th column of A^{-1}, and I_k is the k-th column of the identity matrix I. (The algorithm follows from the definition of the inverse matrix: $A \times A^{-1} = I$). Roughly, how many (long) operations will this good student need to perform? The idea is that the student will try to minimize the number of operations using what he/she learned in this class.

2 Problem II, 20 points.

Experiment with Mathematica's LinearSolve[] function (with the default settings) to determine if it is clever enough to recognize a tri-diagonal matrix T as input, and use a specific algorithm to solve $Tx = b$ system of equations, as opposed to a generic Gaussian elimination. You may want to run a number of jobs of increasing matrix size and see how fast LinearSolve[] handles them. Use `matrix.math` on the class site as a starting point, it contains most of the syntax you will need. Make sure to show the appropriate plots and clearly outline your reasoning.