11-3

CS 53, Fall 2017

Due Nov. 6 at 2:59 pm

Problems 7.9.4, 7.9.5 (using the usual stencil, `Gaussian_Elimination_problems.py`), and the following problem (using the stencil `subfunction.py`)

Hint: For this problem (as for most of the programming problems in this class), you are much better off **not** mutating structures or reassigning to variables.

Problem 1: In this problem you will write code for taking a linear function and finding the "largest" invertible subfunction of that function. This was presented in lecture on October 28 and discussed in lecture on November 7.

Your job is to write a procedure `subfunction(A,V,W)` with the following spec:

The *input* consists of

- an $R \times C$ matrix A over $\mathbb{R}$, represented as a `Mat`,
- a list V of C-vectors over $\mathbb{R}$, and
- a list W of R-vectors.

The input defines a function $f : \mathcal{V} \longrightarrow \mathcal{W}$ where $\mathcal{V} = \text{Span } \{v \ : \ v \in V\}$ and $\mathcal{W} = \text{Span } \{w \ : \ w \in W\}$ and $f(\boldsymbol{x}) = A\boldsymbol{x}$.

The *output* is a pair (V^*, W^*) where

- V^* is a list of vectors that form a *basis* for the domain of f^*, and
- W^* is a list of vectors that form a *basis* for the co-domain of f^*,

where $f^* : \text{Span } (V^*) \to \text{Span } (W^*)$ is the "largest" invertible subfunction of f. By "subfunction", I mean that f^* is defined by the same rule as f, and its domain is a subspace of the domain of f and its co-domain is a subspace of the co-domain of f. By "largest", I mean the dimension of the domain and co-domain are as large as possible.

You may use any procedures that you have written during the course and any modules (including `solve` and `matutils`) that we have provided to you. Note that the solution need not be complicated; my procedure takes four lines, using a subroutine you previously coded.