Lecture 11

Finish up eigenvalues & the power method
Then DONE with simple algorithms!

On to finite terminating algorithms.

For solving a linear system $Ax=b$ we will look at methods that successively eliminate variables from the equations to produce smaller systems. This will let us solve $Ax=b$ in a fixed number of steps.

Then we'll also see how to encode these operations as matrices themselves. This will enable us to store and decompose a linear system into a product of matrices. This is often called the LU or Cholesky decomposition.

We'll see what properties of the matrix this maintains, and also how this will generalize to eliminating multiple equations at once.
function solve1(A::Matrix, b::Vector)
    m,n = size(A)
    @assert(m==n, "the system is not square")
    @assert(n==length(b), "vector b has the wrong length")
    if n==1
        return [b[1]/A[1,1]]
    else
        D = A[2:end,2:end]
        c = A[1,2:end]
        d = A[2:end,1]
        α = A[1,1]
        y = solve1(D-d*c'/α, b[2:end]-b[1]/α*d)
        γ = (b[1] - c'*y)/α
        return pushfirst!(y, γ)
    end
end

function myreduce_all(A::Matrix)
    A = copy(A) # save a copy
    n = size(A,1)
    L = Matrix(1.0I,n,n)
    U = Matrix(1.0I,n,n)
    d = zeros(n)
    for i=1:n-1
        α = A[i,i]
        d[i] = α
        U[i,i+1:end] = A[i,i+1:end]/α
        L[i+1:end,i] = A[i+1:end,i]/α
        A[i+1:end,i+1:end] -= A[i+1:end,i]*A[i,i+1:end]'/α
    end
    d[n] = A[n,n]
    return L,U,d
end