Lecture 16
Analysis of Algorithms

1. The FLOP count of LU with partial pivoting.

2. Two linear systems that Julia can't solve.
   —> Using QR can solve one better, why this is due to the stability of an algorithm.
   —> But no algorithm can solve the other one better, this is due to the conditioning of the problem.

3. The Least squares problem on polynomials and the Hilbert matrix.

4. The formal definition of conditioning and the condition number of a problem.

```julia
function myreduce_all_pivot(A::Matrix)
    A = copy(A)
    for i=1:n-1
        maxval = abs(A[i,i])
        newrow = i
        for j=i+1:n
            if abs(A[j,i]) > maxval
                newrow = j
                maxval = abs(A[j,i])
            end
        end
        if maxval < eps(1.0)
            error("the system is singular")
        end
        j = newrow
        # swap the ith row/column
        tmp = A[i,:]
        A[i,:] .= A[j,:]
        A[j,:] .= tmp
        p[i], p[j] = p[j], p[i]
        L[i,1:i-1], L[j,1:i-1] = L[j,1:i-1], L[i,1:i-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
    end
    d[n] = A[n,n]
    return L, U, d, p
end
```

```julia
A = randn(4,4)
L,U,d,p = myreduce_all_pivot(A)
display(L*Diagonal(d)*U - A[p,:])
```