

```
% MACM 316 - Computing Assignment 3
% Generates different types of random matrices of size N, and tests their
% runtimes
% File name: CA3.m

close all %remove all figures previously generated

vals = [500, 600, 700, 800, 900, 1000, 1250, 1500, 1750, 2000]; %Set values for N

mean_times = double(length(vals)); %Initialize our mean_times vector so it is the same size as vals

% Run through the values of N chosen by me
for i=1:5
    for j=1:length(vals)

        N=vals(j); % Matrix size, chosen from the vector

        % Determine which type of array to test
        if i==1
            % Random matrix
            A=randn(N,N);
        elseif i==2
            % Diagonally dominant
            A=randn(N,N);
            A=A-diag(diag(A));
            A=A+diag(1+sum(abs(A),2));
        elseif i==3
            % Upper triangular
            A=randn(N,N);
            A=triu(A);
        elseif i==4
            % Tridiagonal
            A=diag(randn(N-1,1),-1)+diag(randn(N-1,1),1)+diag(randn(N,1));
        else
            % Tridiagonal sparse
            b1=randn(N,1);
            b2=rand(N,1);
            b3=rand(N,1);
            A=spdiags([b1 b2 b3],[-1:1,N,N]);
        end

        Ntr=500; % Number of trials

        times=zeros(Ntr,1); % Vector of timing data for Ntr trials
        x=ones(N,1); % exact solution vector

        for k=1:Ntr
            b=A*x; % Compute the right-hand side vector
```

```

    %Timing gaussian elimination
    tic;
    z=A\b; % perform gaussian elimination
    times(k)=toc; % record the time taken
end
% Compute the mean of the time
mean_times(j)=mean(times);
end
% Determine which graph to plot
if i==1
    %Plot 1 (random matrix)
    plot(log(vals), log(mean_times), "r*");
    hold on
    p1=polyfit(log(vals), log(mean_times), 1);
    f1=polyval(p1, log(vals));
    plot(log(vals), f1, "b");
    hold off
    title('(1) G.E avg. runtime of NxN random matrix','fontsize',14);
    ylabel('log(Mean runtime)','fontsize',12)
    xlabel('log(N)','fontsize',12)
    %Print the slope of our best fit line to the terminal
    disp("Slope of best fit line for matrix 1: "+p1(1));
    figure;
elseif i==2
    %Plot 2 (random, diagonally dominant matrix)
    plot(log(vals), log(mean_times), "r*");
    hold on
    p1=polyfit(log(vals), log(mean_times), 1);
    f1=polyval(p1, log(vals));
    plot(log(vals), f1, "b");
    hold off
    title('(2) G.E avg. runtime of NxN diagonally dominant matrix','fontsize',14);
    ylabel('log(Mean runtime)','fontsize',12)
    xlabel('log(N)','fontsize',12)
    %Print the slope of our best fit line to the terminal
    disp("Slope of best fit line for matrix 2: "+p1(1));
    figure;
elseif i==3
    %Plot 3 (random, upper triangular matrix)
    plot(log(vals), log(mean_times), "r*");
    hold on
    p1=polyfit(log(vals), log(mean_times), 1);
    f1=polyval(p1, log(vals));
    plot(log(vals), f1, "b");
    hold off
    title('(3) G.E avg. runtime of NxN upper triangular matrix','fontsize',14);
    ylabel('log(Mean runtime)','fontsize',12)
    xlabel('log(N)','fontsize',12)
    %Print the slope of our best fit line to the terminal
    disp("Slope of best fit line for matrix 3: "+p1(1));

```

```
figure;
elseif i==4
    %Plot 4 (tridiagonal matrix)
    plot(log(vals), log(mean_times), "r*");
    hold on
    p1=polyfit(log(vals), log(mean_times), 1);
    f1=polyval(p1, log(vals));
    plot(log(vals), f1, "b");
    hold off
    title('(4) G.E avg. runtime of NxN tridiagonal matrix','fontsize',14);
    ylabel('log(Mean runtime)','fontsize',12)
    xlabel('log(N)','fontsize',12)
    %Print the slope of our best fit line to the terminal
    disp("Slope of best fit line for matrix 4: "+p1(1));
    figure;
else
    %Plot 5 (tridiagonal matrix stored as a sparse array)
    plot(log(vals), log(mean_times), "r*");
    hold on
    p1=polyfit(log(vals), log(mean_times), 1);
    f1=polyval(p1, log(vals));
    plot(log(vals), f1, "b");
    hold off
    title('(5) G.E avg. runtime of NxN sparsed tridiagonal matrix','fontsize',14);
    ylabel('log(Mean runtime)','fontsize',12)
    xlabel('log(N)','fontsize',12)
    %Print the slope of our best fit line to the terminal
    disp("Slope of best fit line for matrix 5: "+p1(1));
end
end
```