Lab #8

Contents

1 Highlighting vertices in a graph

When plotting a graph, it is possible to color certain vertices with another color. This can be achieved with `highlight` function as presented in the following example:

```
clear; close all;

A = [
    0 1 0 1;
    1 0 1 0;
    0 1 0 1;
    1 0 1 0
    ]; % Adjacency matrix

C = [1, 3]; % Vertices to highlight

figure;
G = graph(A);
h = plot(G);
highlight(h, C, 'NodeColor', 'r'); % 'r' for red, 'g'
    for green, 'b' for blue, etc.
```

2 Task #15

- Write a program which creates an edge list representation of an adjacency matrix. Test the program on several graphs from Homework 1.
- In the same program write code which converts the resulting edge list from part 1 of the task back to an adjacency matrix. Check that the original adjacency matrix and the resulting one are the same. Plot the graphs.

Hint: if you want to create a new matrix of the same size as an existing one and fill it with zeros, you can use the following code, where `A` is the existing matrix.

```
new_A = zeros(size(A));
```

3 Task #16

Write a program which for a given subset of vertices I of the graph $G = (V, E)$ checks whether I is an independent set of G. Plot the graph and highlight the vertices which are in the IS.

You can base your code on the following pseudocode that uses an adjancency matrix representation of G:

Algorithm 1: Test for independence

Input : Adjacency matrix A, set of vertices I
Output: Boolean value t (true if I is an independent set and false otherwise)

```
1  Function is_independent_set (A, I)
2      t := true;
3      if length(I) > 1 then
4          for v_index = 1 TO length(I)−1 do
5              v := I[v_index];
6              for w_index = v_index + 1 TO length(I) do
7                  w := I[w_index];
8                  if A[v][w] == 1 then
9                      t := false;
10                 end
11             end
12         end
13     end
14     return t;
15 end
```