Lab 3 and 4

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 Part 1: Test for a vertex cover

Write a program that for a given undirected graph $G = (V, E)$ and a subset of vertices C of V checks whether C is a vertex cover of G. Plot the graph and highlight the vertices of C if C is a vertex cover.

Example of an input is:

```
E = [
    1, 2;
    2, 3;
    4, 5;
    6, 7;
];
C = [1, 2, 3, 4, 5, 6, 7]; % Set of vertices (vertex
   cover)
```

You can use the following pseudocode:

Algorithm 1: Test for a vertex cover

Input : Edge list E, subset of vertices C
Output: Boolean value is_vc (true if C is a vertex cover of G, false otherwise)

```
is_vc = true;
for every edge u, v in E do
    covered = false;
    for j = 1 TO |C| do
        if C[j] == u or C[j] == v then
            covered = true;
        end
    end
    if not covered then
        is_vc = false;
    end
end
```

3 Part 2: Maximal independent set

3.1 Test for independence

Write a program which for a given subset of vertices I of an undirected 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 adjacency matrix representation of G:

Algorithm 2: 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)

```
t = true;
if |I| > 1 then
    for v_index = 1 TO |I| - 1 do
        v = I[v_index];
        for w_index = v_index + 1 TO |I| do
            w = I[w_index];
            if there is an edge between v and w then
                t = false;
            end
        end
    end
end
```

3.2 Maximal independent set

Write a program which for a given graph $G = (V, E)$ finds a maximal Independent Set in it.

You can implement a simple greedy algorithm which is based on the idea of starting from an empty set I, picking vertices of G one by one and adding them to I if I continues to be an independent set.

The algorithm is described in more details in the following pseudocode:

Algorithm 3: Maximal Independent Set

Input : Adjacency matrix A

Output: A subset of vertices I

P = a random permutation of vertices V;
$I = \varnothing$;
for every vertex v *in* P **do**
    $T = I \cup v$;
    if T *is an Independent Set* **then**
        $I = I \cup v$;
    end
end

Hint: To find a random permutation of vertices of G you can use the following command: `P = randperm(size(A, 1));`