

Lab 3 and 4

Contents

1	Highlighting vertices in a graph	1
2	Part 1: Test for a vertex cover	2
3	Part 2: Maximal independent set	3
3.1	Test for independence	3
3.2	Maximal independent set	3

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:

```
1 clear; close all;
2
3 A = [
4     0 1 0 1;
5     1 0 1 0;
6     0 1 0 1;
7     1 0 1 0
8 ]; % Adjacency matrix
9
10 C = [1, 3]; % Vertices to highlight
11
12 figure;
13 G = graph(A);
14 h = plot(G);
15 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:

```
1 E = [  
2     1, 2;  
3     2, 3;  
4     4, 5;  
5     6, 7;  
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)

```
1  $is\_VC = \text{true};$   
2 for every edge  $u, v$  in  $E$  do  
3    $covered = \text{false};$   
4   for  $j = 1$  TO  $|C|$  do  
5     if  $C[j] == u$  or  $C[j] == v$  then  
6        $covered = \text{true};$   
7     end  
8   end  
9   if not  $covered$  then  
10     $is\_VC = \text{false};$   
11  end  
12 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 is_IS (true if I is an independent set and false otherwise)

```
1  $is\_IS = \text{true};$ 
2 if  $|I| > 1$  then
3   for  $v_{index} = 1$  TO  $|I| - 1$  do
4      $v = I[v_{index}];$ 
5     for  $w_{index} = v_{index} + 1$  TO  $|I|$  do
6        $w = I[w_{index}];$ 
7       if there is an edge between  $v$  and  $w$  then
8          $is\_IS = \text{false};$ 
9       end
10    end
11  end
12 end
```

3.2 Maximal independent set

Write a program which for a given undirected 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

```
1  $P$  = a random permutation of vertices  $V$ ;  
2  $I = \emptyset$ ;  
3 for every vertex  $v$  in  $P$  do  
4    $T = I \cup v$ ;  
5   if  $T$  is an Independent Set then  
6      $I = I \cup v$ ;  
7   end  
8 end
```

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