

# Homework 8

Due 06/25/17

June 21, 2017

Answer the following about the *worst-case* analysis for the following graph algorithm:

```
Input:  $G = (V, E)$ : graph with  $n$  vertices and  $m$  edges
Input:  $n, m$ : order and size of  $G$ , respectively
1 Algorithm: GraphMystery
2  $q = \text{Queue}()$ 
3  $visited = \text{Array}(n, n)$ 
4 Initialize  $visited$  to false
5 Enqueue the ordered pair  $(v_0, v_0)$  on  $q$ 
6  $visited[0, 0] = 1$ 
7 while  $q$  not empty do
8    $(v_i, v_j) = q.\text{Dequeue}()$ 
9   for  $v_x \in N(v_i)$  do
10    for  $v_y \in N(v_j)$  do
11      if  $\neg visited[x, y]$  then
12         $q.\text{Enqueue}(v_x, v_y)$ 
13         $visited[x, y] = \text{true}$ 
14      end
15    end
16  end
17 end
18 if all entries in  $visited$  are true then
19   return true
20 else
21   return false
22 end
```

1. What is the time complexity of the nested for loops in one iteration of the outer while loop, when using an adjacency list to store  $G$ ? Your estimate should include the time to call `getNeighbors()` in both lines 9 and 11.
2. What is the total *worst case* time complexity for all iterations of the nested for loops? *Hint:* the worst case behavior involves iterating the outer while loop for each pair of vertices  $(u, v) \in V \times V$ .

3. Find the worst-case time complexity for GraphMystery when using an adjacency list to store the graph. Justify your answer.