DSC 40B - Midterm 02 Review

Problem 1.

The goal of contact tracing is to determine how the spread of a virus occurs. Which type of graph would be best for modelling the spread of a virus?

- ☐ Directed graph
- ☐ Undirected graph

Solution: Directed graph

Problem 2.

A directed graph has 7 nodes. What is the maximum number of edges it can have?

Solution: 49

Problem 3.

An undirected graph has 12 nodes. What is the maximum number of connected components it can have?

Solution: 12

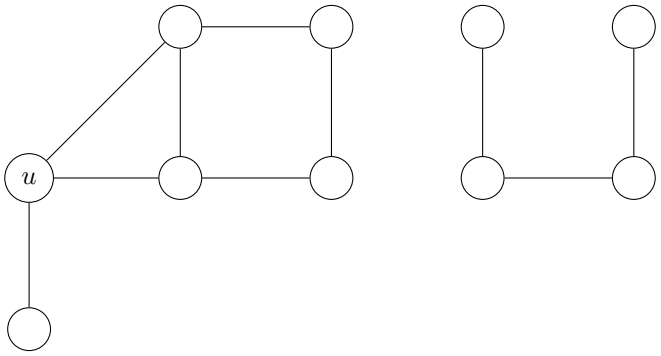
Problem 4.

A directed graph has 5 nodes. What is the largest degree that a node in the graph can possibly have?

Solution: 10

Problem 5.

How many nodes are reachable from node u in the graph?



Solution: 6

Problem 6.

Both BFS and DFS can be used to count the number of connected components in an undirected graph.

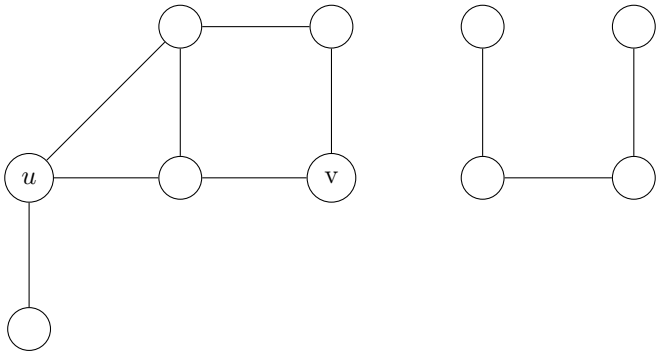
☐ True

☐ False

Solution: True

Problem 7.

How many paths are there from node u to node v in the graph below?



☐ Infinitely many

☐ 4

☐ 3

☐ 5

Solution: Infinitely many

Problem 8.

In an unweighted graph, there is at most one shortest path between any pair of given nodes.

☐ True

☐ False

Solution: False

Problem 9.

An undirected graph has 5 nodes. What is the smallest number of connected components it can have?

Solution: 1

Problem 10.

In a full BFS of a graph G=(V, E), the number of times that something is popped form the queue is 2V if the graph is undirected and V if the graph is directed.

☐ True

☐ False

Solution: False

Problem 11.

In BFS it is possible for the queue to simultaneously contain a node whose distance from the source is 3 and node whose distance from the source is 5.

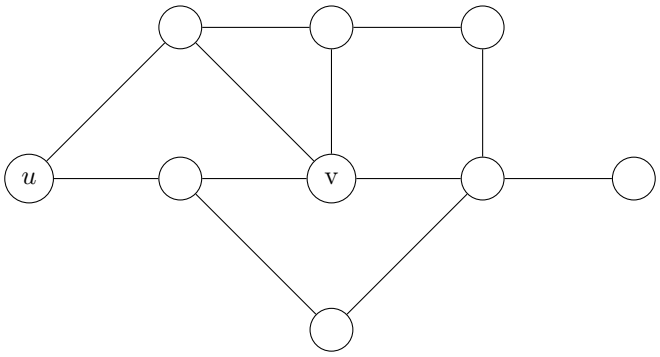
☐ True

☐ False

Solution: False

Problem 12.

Suppose a BFS is run on the graph below with u as the source.



Of course, u is the first node to be popped of the queue. Suppose that node v is the kth node popped from the queue.

a) What is the smallest that k can possibly be?

Solution: 4

b) What is the largest that k can possibly be?

Solution: 5

Problem 13.

Consider the modified mergesort given below:

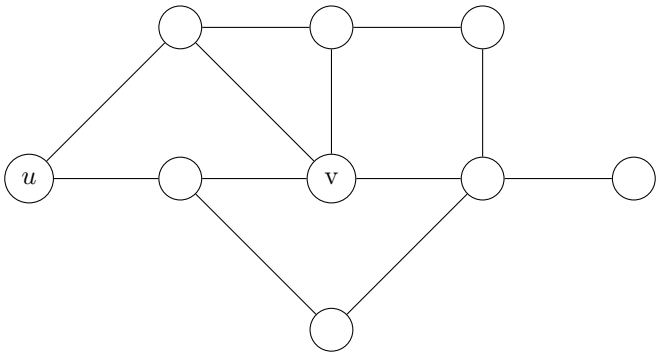
```
def bfs(graph, source, status=None):
    if status is None:
        status = {node: 'undiscovered' for node in graph.nodes}

    status[source] = 'pending'
    pending = deque([source])

    # while there are still pending nodes
    while pending:
        u = pending.popleft()
        for v in graph.neighbors(u):
            # explore edge (u,v)
            if status[v] == 'undiscovered':
                print ("Hey")
                status[v] = 'pending'
                # append to right
                pending.append(v)
        status[u] = 'visited'
```

Suppose this code is run on a connected undirected graph with 12 nodes. Exactly how many times will 'Hey' be printed?

Solution: 11

Problem 14.

Suppose a DFS is run on the graph below with u as the source.



Node u will be the first node marked pending. Suppose that node v is the kth node marked pending.

a) What is the smallest that k can possibly be?

Solution: 3

b) What is the largest that k can possibly be?

Solution: 9

Problem 15.

If DFS is called on a complete graph, the time complexity is $\theta(V^2)$

☐ True

☐ False

Solution: True