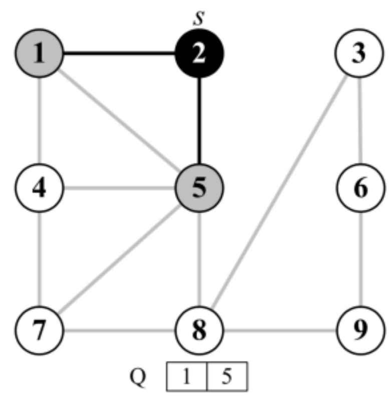
CS 330 – Fall 2020 – Lab 2

Question 0. . BFS refresher, to be completed before lab

You are given the following graph:



Run the BFS algorithm on the above graph from root node s and define the sequence of the nodes visited and the state of the queue as you execute the algorithm (1st step is there).

Question 1. Stacks and Queues

$$E\ A\ S\ *\ Y\ *\ Q\ U\ E\ *\ *\ *\ S\ T\ *\ *\ *\ I\ O\ *\ N\ *\ *\ *$$

(a) A letter means push and an asterisk means pop on the above sequence. Give the sequence of values returned by the pop operations when this sequence of operations is performed on an initially empty LIFO stack.

(b) A letter now means enqueue and an asterisk means dequeue in the given sequence. Give the sequence of values returned by the dequeue operation when this sequence of operations is performed on an initially empty FIFO queue.

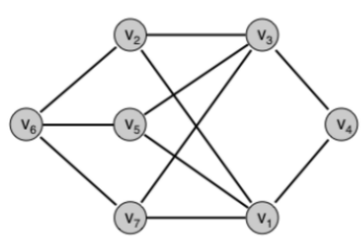
(c) Describe how to implement a queue using two stacks.

Question 2. Complement of a graph

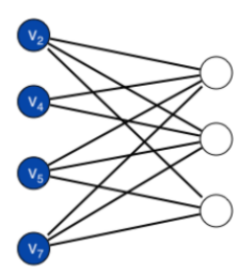
Let $G(V, E)$ be a graph. The complement of G is a graph $G^C(V, E^C)$, such that every edge $(u, v) \in E^C$ if and only if $(u, v) \notin E$. Describe an efficient algorithm for computing G^C given the adjacency-matrix or the adjacency-list of G. Analyze the running times of your algorithms.

Question 3. Deciding bipartiteness

Recall, that a graph $G(U, V, E)$ is bipartite if it has two distinct set of nodes, U and V and edges E, such that any edge has exactly one end in U and one in V. Observe that in the picture below we can see the exact same graph twice. In one of the depictions it is clear that it is bipartite, in the other one it's not.



a bipartite graph G



another drawing of G

In this problem you will apply BFS to find out whether a graph is bipartite.

a) Run BFS on the example above – from a random source – and see how that can help you assign the nodes to the two sets.

b) List three different cycles present in graph G. What do they have in common?

c) Add a new edge (v_1, v_3) to graph G. Note that this creates a few new cycles in the graph, for instance v_3, v_4, v_1. How is this cycle different from the cycles you found in part (b)? Is G still bipartite? Why or why not?

d) Prove that a graph is bipartite iff it does not contain an odd cycle. *Hint: Show that if you run BFS, and there is an edge between two nodes in the same layer, then that graph must contain an odd cycle.*