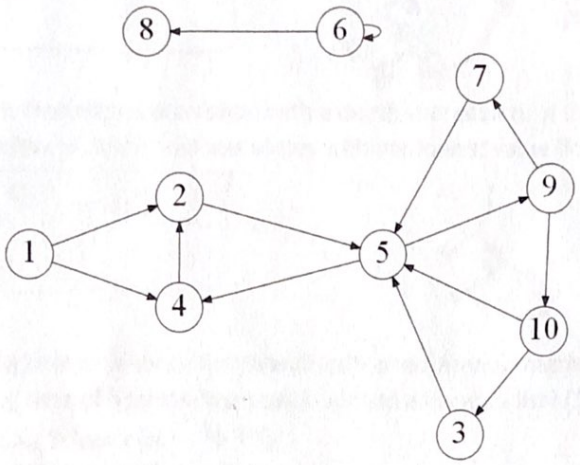
CS 385, Homework 3: Graph Algorithms
"I pledge my honor that I have abided by the Stevens Honor System."

Name: Julia Nelson Date: 03/06/2020

Point values are assigned for each question. Points earned: ____ / 100

Consider the following graph:



1. Draw how the graph would look if represented by an adjacency matrix. You may assume the indexes are from 1 through 10. Indicate 1 if there is an edge from vertex A -> vertex B, and 0 otherwise. (10 points)

indices	1	2	3	4	5	6	7	8	9	10
1	0	1	0	1	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	1	0
6	0	0	0	0	0	1	0	1	0	0
7	0	0	0	0	1	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	0	0	1
10	0	0	1	0	1	0	0	0	0	0

2. Draw how the graph would look if represented by an adjacency list. You may assume the indexes are from 1 through 10. (10 points)

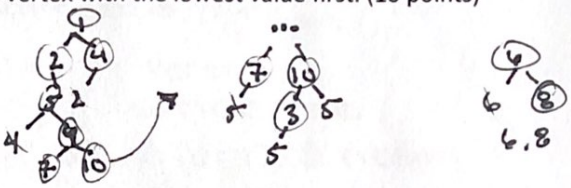
1 → 2 → 4
2 → 5
3 → 5
4 → 2
5 → 4 → 9
6 → 6 → 8
7 → 5
8 →
9 → 7 → 10
10 → 3 → 5

"Pledge" Jwelson

CS 385, Homework 3: Graph Algorithms

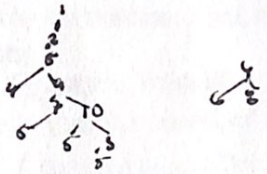
3. List the order in which the vertices are visited with a breadth-first search. If there are multiple vertices adjacent to a given vertex, visit the adjacent vertex with the lowest value first. (10 points)

1, 2, 4, 5, 9, 7, 10, 3
6, 8



4. List the order in which the vertices are visited with a depth-first search. If there are multiple vertices adjacent to a given vertex, visit the adjacent vertex with the lowest value first. (10 points)

1, 2, 5, 4, 9, 7, 10, 3
6, 8



5. a) What is the running time of breadth-first search with an adjacency matrix? (5 points)
   b) What is the running time of breadth-first search with an adjacency list? (5 points)

a) Adjacency Matrix: BFS
checks $(V_i, V_j) \in$ in Edges
$|V|$ = # vertices ∴ $\Theta(1)$ of V·V space
Runtime = $\Theta(V^2)$

b) Adjacency List: BFS
iterates through vertices and # of Edges the next vertex
$|E|$ = # of Edges
Runtime = $\Theta(|V|+|E|)$

6. a) What is the running time of depth-first search with an adjacency matrix? (5 points)
   b) What is the running time of depth-first search with an adjacency list? (5 points)

a) Adjacency Matrix: ~~BFS~~ DFS
check $(V_i, V_j) \in$ Edges
V·V memory
Runtime = $\Theta(V^2)$

b) Adjacency List: DFS
Runtime = $\Theta(|V|+|E|)$

DFS performs better from the differing $|V|$ and $|E|$ values than ~~the~~ higher $|V|$ values in BFS

7. While an adjacency matrix is typically easier to code than an adjacency list, it is not always a better solution. Explain when an adjacency list is a clear winner in the efficiency of your algorithm? (5 points)

Adjacency list is more efficient when # Edges is smaller than the # of vertices because the space efficiency is better than that with a matrix.

"Pledge" [signature]

8. Explain how one can use a breadth-first to determine if an undirected graph contains a cycle. (10 points)

Running BFS on undirected graph, at a vertex, if an adjacent vertex has been visited and is not the parent of the vertex, the graph contains a cycle.

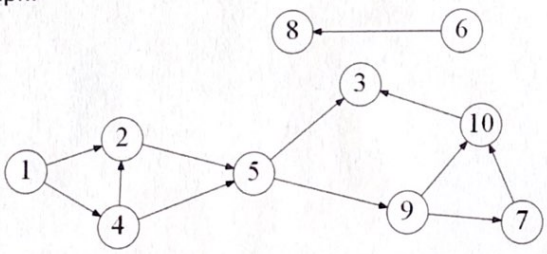
9. On undirected graphs, does either of the two traversals, DFS or BFS, always find a cycle faster than the other? If yes, indicate which of them is better and explain why it is the case; if not, draw two graphs supporting your answer and explain the graphs. (10 points)

Neither BFS or DFS is faster. Both have same adj Matrix/list Runtimes but they depend on the shape of the graph (changing the |V| and |E| values)

10. Explain why a topological sort is not possible on the graph at the very top of this document. (5 points)

The cycles in the graph in question make a Topological sort invalid. When running it on this graph, we first visit the vertex with no "incoming" edges. (vertex ①). After this sorting of vertex ① there are no more vertex with no incoming edges. Thus the sort fails.

Consider the following graph:



11. List the order in which the vertices are visited with a topological sort. Break ties by visiting the vertex with the lowest value first. (10 points)

1, 4, 2, 5, 6, 8, 9, 7, 10, 3