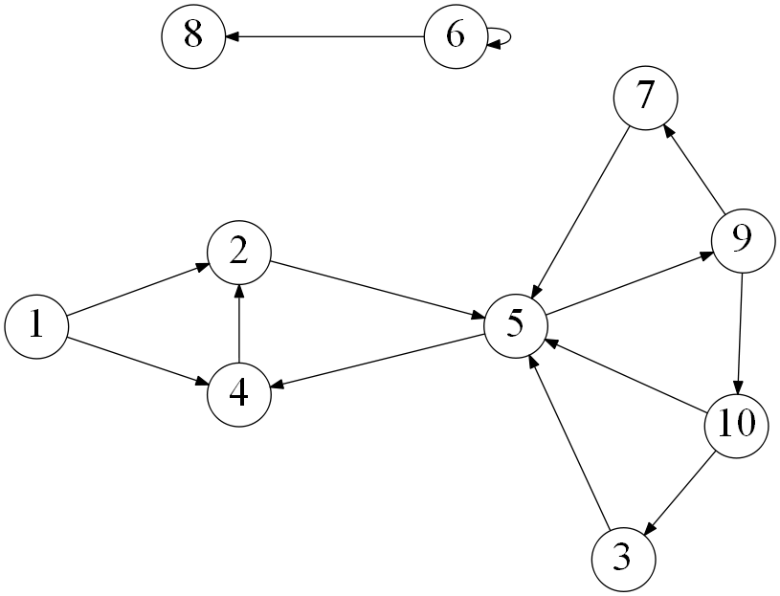
Name: Daniel Detore Date: 10/16/2024

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)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 |  | 1 |  | 1 |  |  |  |  |  |  |
| 2 |  |  |  |  | 1 |  |  |  |  |  |
| 3 |  |  |  |  | 1 |  |  |  |  |  |
| 4 |  | 1 |  |  |  |  |  |  |  |  |
| 5 |  |  |  | 1 |  |  |  |  | 1 |  |
| 6 |  |  |  |  |  | 1 |  | 1 |  |  |
| 7 |  |  |  |  | 1 |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  | 1 |  |  | 1 |
| 10 |  |  | 1 |  | 1 |  |  |  |  |  |

1. 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 | 5, 3 |

1. 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**

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

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

**Θ(V²) for V vertices.**

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

**Θ(V+E) for V vertices and E edges.**

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

**Θ(V²) for V vertices.**

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

**Θ(V + E) for V vertices and E edges.**

1. 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)

**Though its running time is generally slower, an adjacency matrix saves on time as the graph gets more complete; the amount of edges E approaches V², and Θ(V²) with an adjacency matrix is actually better than Θ(V + E) = Θ(V + V²) for an adjacency list.**

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

**If we keep track of every vertex we have visited up until the current vertex, the graph has a cycle if we ever see the same vertex twice and it doesn’t if we don’t.**

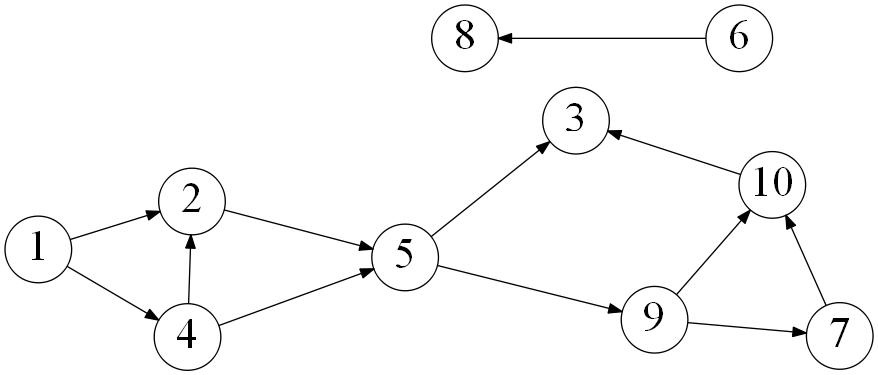
1. On undirected graphs, does either of the two traversals, DFS or BFS, always find a cycle faster than the other? If yes, indicate which of them is better and explain why it is the case; if not, draw two graphs supporting your answer and explain the graphs. (10 points)

**Yes. Breadth-first search is better as it will return to the source vertex as early as possible, while depth-first search will search more vertices before returning to the source vertex.**

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

**It contains at least one cycle, and topological sort cannot be completed if there is a cycle.**

Consider the following graph:



1. 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, 9, 7, 10, 3, 6, 8**