The University of Michigan
Electrical Engineering & Computer Science
EECS 281: Data Structures and Algorithms
Fall 2024

Lab 9: Graphs, Searching Algorithms, and MST Algorithms

Instructions:
Work on this document with your group, then enter the answers on the canvas quiz.

Note:
Be prepared before you meet with your lab group, and read this document so that you know what you must submit for full credit. You can even start it ahead of time and then ask questions during any lab section for help completing it.

You MUST include the following assignment identifier at the top of every file you submit to the autograder as a comment. This includes all source files, header files, and your Makefile (if there is one). If there is not autograder assignment, you may ignore this.

Project Identifier: 472D3C8289DE4915774A47683EC45FFBA373B980

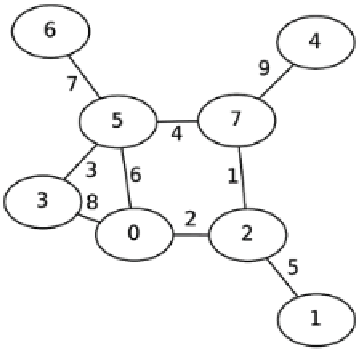
To find the starter code for this lab, go here.

1 Logistics

1. When is Project 4 Due?
   A. 12/04/2024
   B. 12/09/2024
   C. 12/10/2024
   D. 12/13/2024
2. What is the date of the final exam?
   A. 12/12/2024
   B. 12/14/2024
   C. 12/13/2024
   D. 12/16/2024

2 Graphs and Searching Algorithms

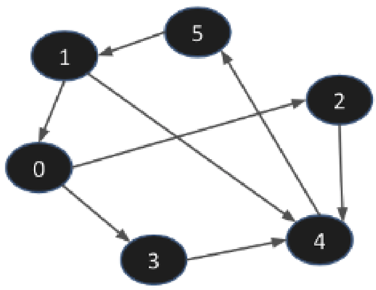
3. Which of the following can be used to describe the graph below? Select all that apply.



   A. Unweighted
   B. Directed
   C. Undirected
   D. Weighted
   E. Bipartite

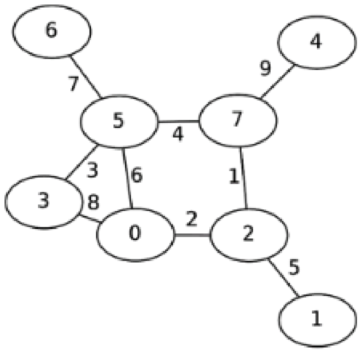
4. Select all the graphs that are sparse below:
   A. A fully-connected graph
   B. A friend network (eg. Facebook – people are verticies and friendships are edges)
   C. Friend group chat (Where people are verticies and friendships are edges)
   D. The internet (Where webpages are vertices and hyperlinks are edges)
   E. Campus transit system (Where buildings on campus are vertices and bus routes are edges)

5. On the directed graph below, select the nodes with the correct adjacency list. Select all that apply:



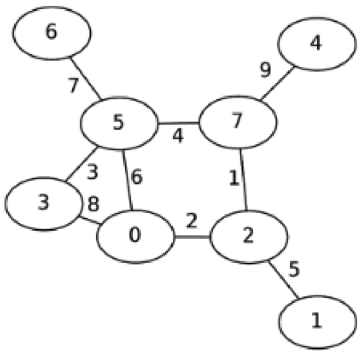
   A. Node 0: [2, 3]
   B. Node 1: [0, 4, 5]
   C. Node 2: [4]
   D. Node 4: [1, 2, 3, 5]
   E. Node 5: [1]

6. What is the sequence of verticies added to the MST for the graph below when running Prim's algorithm, if you start with vertex 7?



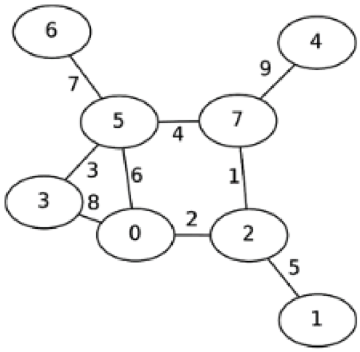
A. 7,2,0,5,1,3,4,6
B. 7,2,0,5,3,1,6,4
C. 7,2,0,3,5,1,6,4
D. 7,2,5,4,0,1,3,6
E. 7,2,5,4,0,3,1,6

7. What is the sequence of verticies added to the MST for the graph below when running Prim's algorithm, if you start with vertex 0?



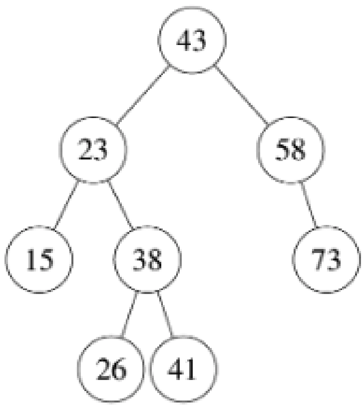
A. 0,2,7,5,3,1,6,4
B. 0,2,7,5,1,3,6,4
C. 0,7,2,3,5,1,6,4
D. 0,2,1,7,5,3,6,4
E. 0,7,2,5,3,1,6,4

8. What is the sequence of edges added to the MST for the graph below when running Kruskal's algorithm? Use the edges' weights as their label for this question.



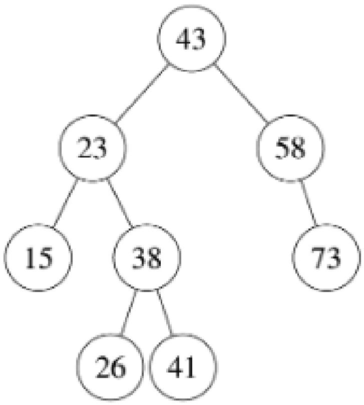
A. 1,4,9,2,3,5,7
B. 1,2,4,5,3,9,7
C. 1,2,4,3,5,7,9
D. 1,2,3,4,5,7,9
E. 1,2,3,4,5,6,7,8,9

9. What is the order of transversal on the tree below using a BFS algorithm with adding the left child to the data structure first?



A. 43,23,58,15,38,73,26,41
B. 43,15,23,38,26,41,58,73
C. 43,23,58,15,38,26,41,73
D. 43,58,23,73,38,15,41,26

10. What is the order of transversal on the tree below using a DFS algorithm with adding the left child to the data structure first?



A. 43,23,58,15,38,,73,26,41
B. 43,58,73,23,38,41,15,26
C. 43,58,23,15,38,26,41,73
D. 43,58,73,23,38,41,26,15

11. Select the statement that is false.
A. Both BFS and DFS will alsways return the shortest path to a target vertex in an unweighted graph.
B. BFS has first-in first-out (FIFO) ordering, while DFS has last-in first-out (LIFO) ordering.
C. Both BFS and DFS take $O(|V| + |E|)$ time.
D. If the target vertex isn't found, BFS and DFS will visit every vertex in a connected graph.
E. DFS and BFS do not work well on weighted graphs to find the shortest path to a target vertex.

12. You currently have a graph that represents several airports (vertices) and the flights that connect these airports (edges), weighted by the distance of each flight. You decide to implement this graph using an adjacency list. Let V represent the number of vertices in this graph, and let E represent the number of edges. If you are given an airport x, what is the worst-case time complexity of determining if any flights depart from airport X?
A. $\theta(1)$
B. $\theta(E)$
C. $\theta(V)$
D. $\theta(1 + \frac{E}{V})$
E. $\theta(V^2)$

13. Which of the following statements is/are **true**? Select all that apply.
    A. A stack is a good data structure to use for a depth-first search (DFS)
    B. A stack is a good data structure to use for a breadth-first search (BFS)
    C. A stack is a good data structure to use for conducting a level-order traversal of a binary tree
    D. A queue is a good data structure to use for a depth-first search (DFS)
    E. A queue is a good data structure to use for a breadth-first search (BFS)
    F. A queue is a good data structure to use for conducting a level-order traversal of a binary tree
14. Suppose you have a binary tree of height 281, where leaf nodes have height 1, and you want to search for an element k. You know that k exists as a distinct element in this tree, and that it is a leaf node. Which of the following statements is/are true? Select all that apply.
    A. If you conduct a depth-first search for k, you'll never have to store more than 281 nodes in memory.
    B. If you conduct a breadth-first search for k, you'll never have to store more than 281 nodes in memory.
    C. If you are concerned about the memory efficiency of the search, you should use a queue to conduct this search instead of a stack
    D. the path from the root to the element k returned by a breadth-first search will always be shorter than the path returned by a depth-first search.
    E. While a a breadth-first search will always find the element k, it is possible that a depth-first search may fail to find k since depth-first searches do not always visit every element

3 Handwritten Problem

This problem is to be submitted independently. We recommend trying it on your own, checking your answer with your group and discussing solutions, and then submitting it at the end of lab. These will be graded on completion, not by correctness. However, we want to see that you were thinking about the problem. Please implement your solution in `diameter.cpp`. The header file for this problem is `diameter.h` The starter files can be found on Canvas.

Your task: Given an **undirected** connected graph, check if the graph contains a cycle. Implement the following function:

```
bool is_graph_cyclic(const std::vector<std::vector<int>> &adj_list);
```

Feel free to use a helper function!

You can assume that if u and v are adjacent, then adj_list[u] will contain v and adj_list[v] will contain u. Also, u will not appear in adj_list[u].

Complexity: O(V + E) time

