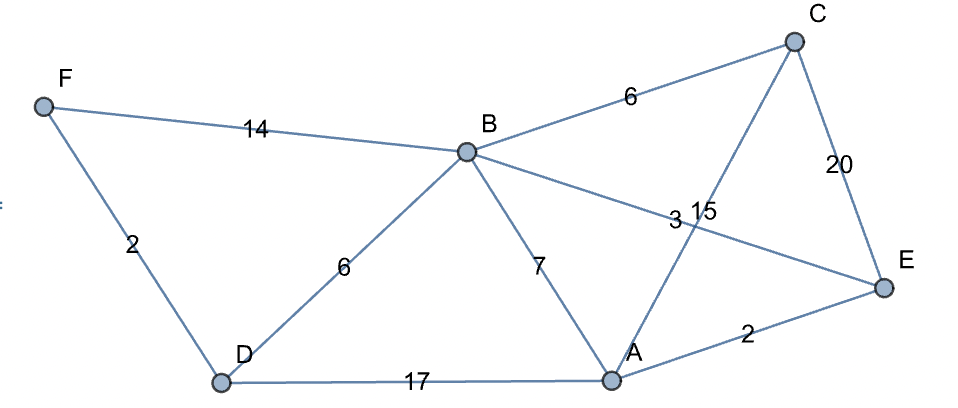
MTH 325 Fall 2024 – Exam 3, Skills 9—12

Instructions: This packet contains problems for the newly acquired Skills 9—12. **If you need to retake any of Skills 1—8 you can find those printed separately**. Complete your work in the spaces provided below each set of Skill questions. Reminder: unless otherwise stated, you are required to provide not only correct answers but clear explanations for each response.

**Skill 9: I can use Dijkstra's Algorithm to find a minimum distance spanning tree for a weighted graph.**

Consider the weighted graph below:



Note, the distance from D to B is 6, and the distance from B to C is 6.

Implement Dijkstra’s Algorithm to find the shortest paths from vertex F (upper left of the graph) to all other vertices (A, B, C, D, and E) in the graph. Your work should consist of two things: A list of visited vertices given in the order that they are visited in the algorithm, and a table showing the distances from F to the other vertices with the updates to distances specified in Dijkstra’s Algorithm.

**Skill 10: I can execute a breadth-first and depth-first search in a graph.**

Consider the graph G below:



1. Execute a depth-first search on G starting at vertex 2. Your final submission should consist of two things: a list of visited vertices in the order in which they are visited, and a history of the stack or queue used to implement the search. Use numerical ordering (low to high) to add vertices into the stack or queue.
2. Repeat part 1 except use a breadth-first search.

**Skill 11 (CORE): I can represent a directed graph in different ways, and determine information about a graph using different representations.**

Consider the directed graph *G*:



1. State the adjacency matrix for G. Assume that the rows and columns correspond to vertices 1, 2, 3, 4, 5, and 6 in that order.
2. State the Python dictionary for G. Be sure to use correct Python syntax for dictionaries, for example {a : [b,c]}.
3. List the in- and out-degrees of each vertex, using the table below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| v | 1 | 2 | 3 | 4 | 5 | 6 |
| In-degree |  |  |  |  |  |  |
| Out-degree |  |  |  |  |  |  |

**Skill 12: I can use Warshall’s algorithm to construct the transitive closure of a directed graph.**

Consider the directed graph whose adjacency matrix is:

A number of black numbers

Description automatically generated with medium confidence

Assume that the vertices are 0, 1, 2, 3 and that the rows and columns correspond to those vertices in this order.

1. Complete the table below for the first six steps of Warshall’s algorithm. The first two rows have the values for the counters in the loops filled in for you. The other values of the loop counters must be done in the correct order given by the pseudocode for Warshall’s algorithm given in class.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| k | i | j | W[i,j] | W[i,k] | W[k,j] | Result |
| 0 | 0 | 0 |  |  |  |  |
| 0 | 0 | 1 |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. Complete the step in Warshall’s algorithm correspoinding to k = 0, i = 3, and j = 1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| k | i | j | W[i,j] | W[i,k] | W[k,j] | Result |
| 0 | 3 | 1 |  |  |  |  |