



## Lab 5: Graph Traversals

Due: Before next week's exam

You will be assigned a partner to work with on this lab. Only one submission per group is needed.

Group members: \_\_\_\_\_

Learning goals:

1. To understand basics of the breadth-first and depth-first graph traversal algorithms.
2. To understand how an implementation of each works.
3. To learn how to perform each of these traversals manually on a small sample graph.

---

### Submitting

Once all written items are initialed to indicate completion, turn in one copy of this handout. Be sure names of all group members are clearly on the first page.

---

### Grading

Score:  / 100

We begin by exploring the ideas of graph traversals, and the differences between the breadth-first and depth-first disciplines, using METAL's visualizations of these algorithms.

The idea of a graph traversal is that, given a graph and a starting vertex, come up with a order in which vertices are visited such that each vertex visited is one that can be reached directly by an edge from a vertex that has already been visited.

In your favorite standards-compliant browser, visit METAL's Highway Data Examiner (<http://courses.teresco.org/metal/hdx/>).

**? Question 1:**

In HDX, load up the "Delaware (State)" graph. For "Algorithm to Visualize" choose "Graph Traversal/Connected Components". For this first run, leave the "Order" as "Breadth First". For the "Start Vertex", choose the vertex in the far southeast corner of the state (which should have label DE1/MD528@MD/DE). Start the algorithm and observe the progress as the vertices are visited and become blue. Show your instructor the completed visualization. (1 point)

**? Question 2:**

Reload HDX and the same graph. This time, for "Order", choose "Depth First". Again, start the algorithm and observe the progress as the vertices are visited and become blue. Show your instructor the completed visualization. (1 point)

**? Question 3:**

Describe in a few sentences your first impressions of the order in which vertices are visited using these two traversal disciplines. (5 points)

**? Question 4:**

How many vertices remain “Undiscovered”, *i.e.* not visited by the traversal, when you completed your traversals of the Delaware graph? Why? (5 points)

**? Question 5:**

Traversal algorithms can also be used to count and/or identify the *connected components* of a graph. In HDX, this functionality is enabled by selecting the “Find all connected components” toggle in the “Graph Traversal/Connected Components” visualization. Select that and re-run the visualization for the Delaware graph. Show your instructor the completed visualization. (1 point)

**? Question 6:**

Does it matter which start vertex and which traversal discipline you choose when finding connected components? Explain briefly. (3 points)

**? Question 7:**

Use HDX to find the number of connected components in the “Hawaii (State)” graph. (This is a “Route Within a Single Region” and has about 450 vertices, in case you’d like to use the filters to find it more easily.) Write your answer below and show your instructor the completed visualization. (3 points)

**? Question 8:**

Is there any single island that has multiple connected components in METAL’s data? If so, where? (3 points)

Next, we consider in more detail how the algorithms work. Reload HDX and load up the “Aruba (Constituent Country)” graph. Select the “Breath First” traversal, select the point `Rt1A/Rt1B@SerCol` as the starting point, which is southeasternmost point. Set the speed to “Painfully Slow” and start the visualization, but pause it right away, before the first step of the algorithm is complete. You’ll know you stopped it soon enough if the yellow “Visiting” line is still referring to `Rt1A/Rt1B@SerCol` and that vertex is shown in yellow on the map.

As the algorithm proceeds, the vertices and edges are added to a *spanning tree* that consists of the vertices that have been visited and the edges that were used to connect them with the previously visited vertices.

### ? Question 9:

The vertices and edges shown in green are those that have been “discovered” and are candidates for the next to be included in the spanning tree. What data structure is being used for this purpose with the breadth-first traversal? (2 points)

### ? Question 10:

Which vertices are in the list of discovered vertices when visiting the start vertex? You may give vertex numbers rather than writing out the names. (2 points)

### ? Question 11:

Resume the simulation for one iteration, then pause it again. Now what is in the list of discovered vertices? (2 points)

### ? Question 12:

Resume and pause again after one more iteration. Now what is in the list of discovered vertices? (2 points)

### ? Question 13:

Resume and pause again after one more iterations, when the vertex labeled `Rt1A@Rt4_E&Rt1B@Rt4_E&Rt4@Rt1_S` is being visited. Now what is in the list of discovered vertices? (3 points)

**? Question 14:**

Resume and pause again after two more iterations, when the two vertices you wrote down for the previous question have been visited. Now what is in the list of discovered vertices? (3 points)

**? Question 15:**

At this point in the simulation, how many edges need to be traversed along the shortest path, defined as fewest number of edges traversed, from the start vertex to each of the two that were just visited? (2 points)

**? Question 16:**

At this point in the simulation, how many edges need to be traversed along the shortest path, again by number of edges traversed, to get from the start vertex to the ones currently in the list of discovered vertices? (2 points)

**? Question 17:**

Can you find any vertices that are not yet visited or in the list of discovered vertices which can be found by traversing fewer edges than your response to the previous question? (2 points)

**? Question 18:**

Resume and pause again after five more iterations, when the algorithm is visiting the vertex labeled  $R_{t+1B}/R_{t+7}$ . Until this point, all edges from each newly visited vertex have been colored green. One this time has been colored pink, meaning that it was “discarded on discovery”. Why does the algorithm discard this edge and not add the vertex at its other endpoint to the list of discovered vertices? (3 points)

**? Question 19:**

Resume and pause again after one more iteration, when the algorithm is again visiting the vertex  $R_{t+1B}/R_{t+7}$ . If it was already visited, why is the algorithm visiting it again? (2 points)

**? Question 20:**

The pink edge should now also be grey, indicating that it is “discarded on removal” instead of being added to the spanning tree (in which case it would become blue). Why do we not need to add this edge to the spanning tree? (3 points)

**? Question 21:**

As you watch the remainder of the simulation, notice that edges that are not ultimately part of the spanning tree are first colored pink as “discarded on discovery” and later grey as “discarded on removal”. Why does each such edge get considered twice, and what condition causes the algorithm to discard on discovery and what condition causes the algorithm to discard on removal? (3 points)

**? Question 22:**

When the algorithm completes, what is the mathematical relationship between the number of vertices and the number of edges in the spanning tree? Would this always be the case for any graph? Explain briefly. (5 points)

**? Question 23:**

Next, load up the Aruba graph again but this time select the traversal discipline to be “Depth First”. What data structure is now being used for the list of discovered vertices? (2 points)

**? Question 24:**

Start a depth-first visualization of the Aruba graph and pause it after 8 iterations, when the algorithm is visiting the  $R_t 4 / R_t 6$  vertex. What vertices are in the list of discovered vertices? (2 points)

**? Question 25:**

There exists a shortest path, in terms of the number of edges traversed, from the starting vertex to any other vertex (not necessarily using edges in the spanning tree). At this point in the simulation, what is the length of the longest among these shortest paths to any vertex currently in the spanning tree? (4 points)

**? Question 26:**

At this point in the simulation, what is the length of the longest among the shortest paths to any vertex in the list of discovered vertices? (4 points)

**? Question 27:**

Load the “Siena College (25 mi radius)” graph into HDX and run both Depth First and Breadth First visualizations at a faster speed, paying attention to the size of the list of discovered vertices as each proceeds. Which traversal discipline results in a larger list? Explain briefly why this would be. (5 points)

Read Levitin Section 3.5.

**? Question 28:**

| Levitin Exercise 3.5.1, p. 128. (10 points)

**? Question 29:**

| Levitin Exercise 3.5.4, p. 128. (10 points)

**? Question 30:**

Please give any suggestions to improve the visualizations of the graph traversals in METAL's HDX that you believe would make it easier to use and/or a more effective teaching tool. (5 points)