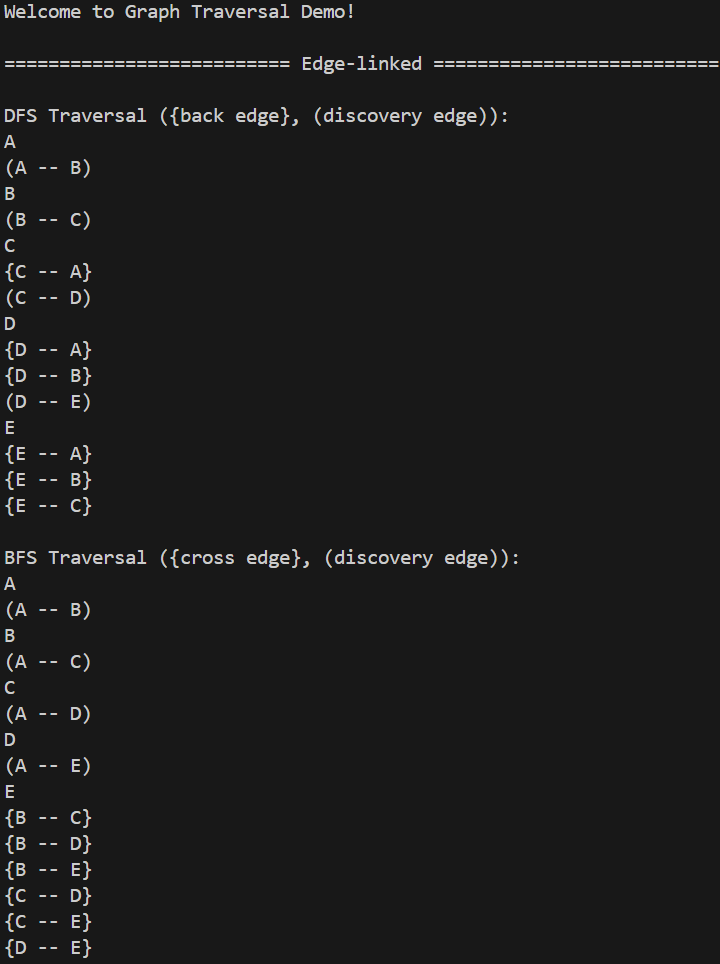
Submit a writeup, screenshots of the results, and code files in a zip file.A white paper with black writing on it

Description automatically generated

**Figure 1.** Hand calculation for order of vertex and edge discovery for DFS and BFS traversals of a complete graph with 5 vertices, with back/cross edges indicated.



**Figure 2.** Results for edge list graph representation.

A screenshot of a computer program

Description automatically generated

**Figure 3.** Results for adjacency list graph representation.

A screenshot of a computer program

Description automatically generated

**Figure 4.** Results for adjacency matrix graph representation.

A traversal of a complete graph with 5 vertices should visit all 5 vertices and all 10 edges. The implementation of both traversals prioritizes the next visit based on the order of the vertices value in its respective data type (lowest alphabetical value, lowest integer, etc.). By this property, between discovery vertices, the depth-first search discovers an increasing number of back edges each time we search deeper in the graph (1 back edge after visiting C, 2 back edges after visiting D, and 3 back edges after visiting E). The bread-first search, due to the graph being complete (all vertices have an edge to all other vertices), discovers all vertices in the graph sooner than does DFS. After exploring all neighbors of the start vertex, the BFS implementation moves on to the next vertex based on it’s element value (A to B to C and so forth). Since all vertices are discovered at this point, the remaining edges are cross edges.

A goal with this portion of the lab was to use a single BFS/DFS implementation which was compatible with all three graph implementations. This proved challenging but made for less code in total.