1. **The worst-case ART of adjacentVertices is O(V), because it uses hash sets and/or hash maps, which are O(1) operations.**

**The worst-case ART of edgeCost is O(1) because the operations it uses, such as checkVertexExists and looking up edges in edgesMap, are O(1) operations.**

**The worst-case ART of shortestPath is O((V + E) log V). Initializing a priority queue and hash map for augmented vertices takes O(V) time. Removing an element from or updating said priority queue is an O(log V) operation. Finally, reconstructing the path takes O(V) time in the worst case. We are left with an O(V log V) worst-case ART for processing all vertices, and O(E log V) for processing all edges.**

**Thus, the total time complexity is O((V + E) log V).**

1. **To test my code, I utilized the provided vertex.txt and edge.txt files. I plugged in random vertices and observed the output. I also tried entering invalid values or the same vertex twice to ensure the program responded accordingly.**