**CSE 373 Homework 5 Write up**

**Ryan McDaniel 1026006**

**Alyanna Castillo 0962947**

1. **Describe the worst-case asymptotic running times of your methods** adjacentVertices**,** edgeCost**, and** shortestPath**. In your answers, use |E| for the number of edges and |V| for the number of vertices. *Explain and justify your answers.***

**adjacentVertices() has a cost of O(1), since that is the cost of the get() method of a Java HashMap.**

**edgeCost is O(|E|), because in the worst case all edges must be checked.**

**shortestPath is O(|V|^2), because in the worse case the dijkstra helper function within it goes through 2 loops – a while loop and within that a for-each loop.**

1. **Describe how you tested your code.**

**For both parts, we merely ran the provided text files and tested various cases for the distances. For example, ATL to SEA and SEA to ATL should be the same, etc.**

**In part 2, the functionality was verified with a test file that listed the start vertex, end vertex, and the cost between the paths. The test file followed the format:**

**A**

**B**

**10**

**B**

**A**

**5**

**B**

**C**

**20**

**A**

**C**

**10**

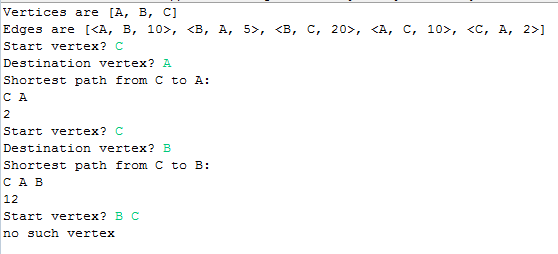
**C**

**A**

**2**

**By using this test file for edges, we were able to test the shortest path function easily. The cost to go from A to C, the options could be (A to B) + (B to C) vs (A to C) = 10 + 20 vs 10 = 30 vs 10. If the shortest path function picked the path that cost 10 for A to C, then we would know our algorithm works.**

**Below is a screenshot of our testing session:**

****

1. **If you worked with a partner:   
   a) Describe the process you used for developing and testing your code. If you divided it, describe that. If you did everything together, describe the actual process used (eg. how long you talked about what, what order you wrote and tested, and how long it took).**

**We divided the work so that one person did part 1 and the other person did part 2. Part 1 acted as a foundation for part 2 – constructors were implemented along with get functions for private fields. With part 2, private helper functions were built for the shortestPath function to calculate the shortest distance and use dijkstra’s algorithm. New fields were also added in Vertex to accommodate dijkstra’s algorithm (path and distance).**

**Debugging / testing was done together so everyone knew what the other person was doing. In addition to the provided text files, our own text files were used to make it easier to compare results (as described above).**

**b) Describe each group member's contributions/responsibilities in the project.**

**Ryan:**

**Part 1 (Constructors, get functions for private variables, print out), testing files**

**Alyanna: Part 2 (ShortestPath, smallestDist, dijkstra), README document**

**We both had to debug each other’s codes together and modify parts of it (i.e. adding an extra field to vertex, etc.)**

**c) Describe at least one good thing and one bad thing about the process of working together.**

**Good: Work division.**

**Bad: Keeping consistent with each other’s variable names and clashes in GitHub syncing.**

1. **If you did any above-and-beyond, describe what you did.**

**We did nothing extra.**

**Appendix**

Place anything that you want to add here.