

README.pdf  
Josephine Lee  
leejosephines@wustl.edu  
CSE241 lab 5

**Implementation:**

In my min-first PriorityQueue, the “nodes” are the Element class, and these each have a Handle. I implemented a swap method to make things easier, and this method made sure to swap all the data in both the Element and the Handle.

I also implemented the methods bubbleUp and bubbleDown (i.e. heapify) as described in class, to use in the methods insert, decreaseKey, and extractMin, which were also implemented as described in class.

For ShortestPaths, I implemented Dijkstra's algorithm as described in class, and kept track of the parent node by keeping a fromEdge array of parent edges.

For the extra credit portion, I added on the layover time to the weight of the edge as the total “weight” to consider when calling decreaseKey, and for updating the distances when decreaseKey was successful. The formula for layover time given in the instructions was very useful, and I simply used that. For calculating a layover, the beginning time for the first airport is simply the startTime. For other airports, it is the endTime of the parent edge in the fromEdge array. In hindsight, it was helpful that I stored an array of parent edges rather than an array of parent vertices or Elements.