PriorityQueue:

Here, I use a class Element<T> to store the T type value with key and handle, where handle is used to store the value of index of the element. An ArrayList is used to store all the elements as an array. I implement insert, extractMin, min and decreaseKey functions and keep the first slot null as we discussed in the class. On the other hand, I implement the handleGetKey and handleGetValue functions to help the find the shortest path.

ShortestPaths:

I use class Node to store the vertex, distance, parent, and handle, edge value, and initialize the parent null, distance infinite. After that, it inserts all the nodes into PriorityQueue. Then I implement Dijkstra to find all the shortest path from start node to every node.

In the returnPath, it can output the shortest path from start node to end node by read the parent value of every node. And print them as string with the time spent on every travel.

Handle:

In this class, I store the index of each Element<T> used in ArrayList, and add this value in to the Element<T> class.