Big Oh

Dijkstra

*O*((*n* + *m*) log *n*)

Using the priority queue implementation, we have to implement a way to change the keys as we change the weights. Since we know where to place it will take O(log n) time to do key updates. We then have to go through each vertex and its edges which adds the O(n+m). In the end its *O*((*n* + *m*) log *n*).

MST

*O*((*n* + *m*) log *n*)

Using the priority queue implementation, we have to extract each vertex in each iteration which takes O(log n) time. In addition, we can update each cost value in O(log n) which is considered at most once for each edge. The other stages of the algorithm are O(1) time so this algorithm ends out being O((n+m) log n).