Evan Zhang | z5383657

Question 3

3.1, 3.2, 3.3

Add each vertex's starting cost and finishing cost together in a separate array taking O(n) time. Find the minimum of this array taking O(n) time. This is likely the smallest score of any path in the graph.

We will use the Dijkstra's algorithm using an augmented heap outlined in the lecture, with a few minor changes:

- In the beginning, $d_s = s_s + f_s$ and is added to a separate array in a tuple with 's' as the starting and finishing vertex
- Instead of deleting the distance from the min-heap, we put it into the separate array in a tuple along with start and end vertices.
    - When we do this, we will also add the finishing cost of the end vertex.
- will run it on each vertex in the given graph

We search the n^2 size list for the smallest distance, taking $O(n^2)$ time.

Since the algorithm runs in O(mlogn) time on n vertices, the algorithm's time will be dominated by O(mnlogn) time. Since m ≤ (n-1) as the graph is simple, this can be rewritten as $O(m^2 \log n)$.

Now, $O((n+m)^2\log(n+m))$ is equivalent to $O(m^2\log(n+m))$ for the same reason and the log can be simplified to $O(m^2 \log n)$.