Team JPEG

Scrum Master: Jesus Sanchez

Product Owner: Sabrina Dang

Members: Agnes Ohide, Jason Zhang, Lee Evans

Data Structures Used

The data structures used for this project include: vectors, unordered maps, and graphs.

Big-O Analysis

1. Shortest Paths (Dijkstra’s Algorithm)
   1. The shortest paths algorithm utilizes Dijkstra’s method in order to return the shortest path to every vertex from an indicated vertex. If the graph object has previously called any of the traversal functions, the vertices and discovery edges must be marked as unvisited and unused which calls a reset function that runs in O(n2) time. After the starting vertex has been added to T, a while loop is used that runs until all vertices have been marked as visited, adding another O(n) to the runtime of this algorithm. Inside the while loop, the function findClosest() is called for to find the closest vertex from the vertices stored in T. This function runs in O(n^3) time since it contains a while loop that runs for the size of T to find the shortest distance from the start from all of the unvisited edges in T. Inside this while loop, the functions edgesDiscovered(), distanceFromStart(), and distanceBetween() are all called twice and all run in O(n) time adding O(6n) to the runtime of findClosest(). Furthermore, the function smallestEdge() that is also called in the loop twice runs in O(n2) time which leads to an overall runtime of O(6n) + O(2n2). So, because this is encompassed within the while loop, findClosest() runs in O(6n2) + O(2n3) or O(n3). Now, coming back to the shortest paths algorithm, because the findClosest() function is called n times by the while loop, the runtime of this while loop is O(n4). Finally, taking all of this into consideration, the overall runtime of the shortest paths algorithm is findClosest() O(n4) + O(n2) + O(n) or O(n4).
   2. **Notes:**
      1. findVertex() = O(n)
      2. distanceBetween() = O(n)
      3. edgesDiscovered() = O(n)
      4. distanceFromStart() = O(n)
      5. smallestEdge() = O(n^2)
2. Prim-Jarnik’s Algorithm
3. Shortest Trip Algorithm
4. Depth-First Search Algorithm
5. Breadth-First Search Algorithm