1. Dijkstra and AStarSearch only differ substantially in the number of nodes they explore in the process of finding a route. As you've already completed those methods, Let's look at this in more detail.

For this question, uncomment (or add) the following code to the **main method in MapGraph.java**. If you don't already count the number of items removed from the queue, you should add that to your code now. (Whenever an element is removed from the priority queue, you increment the count, then print out the count before the method returns. Elements are only added to the queue only when you find a "lower-cost" path to that element.)

MapGraph theMap = new MapGraph();

System.out.print("DONE. \nLoading the map...");

GraphLoader.loadRoadMap("data/maps/utc.map", theMap);

System.out.println("DONE.");

GeographicPoint start = new GeographicPoint(32.8648772, -117.2254046);

GeographicPoint end = new GeographicPoint(32.8660691, -117.217393);

List<GeographicPoint> route = theMap.dijkstra(start,end);

List<GeographicPoint> route2 = theMap.aStarSearch(start,end);

How many nodes were visited by Dijkstra and AStarSearch (if you are close but not exact to one of the results below, chose the one closest to your result)?

Dijkstra: 65, AStarSearch: 13

Dijkstra: 97, AStarSearch: 25

Dijkstra: 97, AStarSearch: 13

Dijkstra: 82, AStarSearch: 25

Dijkstra: 65, AStarSearch: 19

Dijkstra: 97, AStarSearch: 19

Dijkstra: 65, AStarSearch: 25

Dijkstra: 82, AStarSearch: 13

Dijkstra: 82, AStarSearch: 19