

- What data structure(s) will you represent the data in the file “attractions.csv”?

*A linked list. Because of its dynamic data structure, it can grow and shrink at runtime, as compared to modifying arrays. It also has minimal memory wastage.*

```
LinkedList<String> attractions = new LinkedList<String>();
```

- What data structure(s) will you represent the data in the file “roads.csv”?

*A custom defined class, Graph, will be used to store and represent this data. This class is documented in the UML diagram attached.*

```
Graph graph = new Graph();
```

- What algorithm(s) will you use to find the shortest route from starting city to ending city through all the specified events?

*Ideally, Dijkstra's algorithm is easier to find the shortest route between any two cities. The Shortest Path First algorithm, SPF algorithm is relatively easy to understand and implement.*

*My approach is the problem to be solved into several sub-problems. It solves a series of sub-problems, each of which may have several possible routes and where the route selected for one city may have an effect on the possible routes of subsequent cities. A start city is chosen and then bound to a very large value. Select the cheapest route between the unvisited and current city and then add the distance to the current distance. Repeat the process while the current distance is less than the bound. If the current distance is less than the bound, we're done. Add up the distance so that the bound distance will be equal to the current distance. Repeat this process until all the routes have been covered.*