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

Used a linked list. Because of dynamic memory allocation can effectively allocate and utilize memory resources.

```
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.

```
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?**

In generally, Dijkstra algorithm is more suitable for finding the shortest route between two cities. The shortest path and SPF are easy to understand.

The method I adopt is to decompose into subproblems, each subproblem may have several routes, and the route chosen for a city may have an impact on the possible routes.

Select a starting city and bind to a larger value. Select the cheapest route between the current and not visited cities, then add the distance to the current distance.

When the distance is less than the limit. If the current distance is less than the limit, we're OK.

Add up the distance and the binding distance will be equal to the current distance.

Repeat this process until all routes are covered.