Data Structures Used:

1. Vectors
2. Arrays
3. Lists
4. String List
5. Unordered Map

Big (O) Discussion:

void filteredTeams::populateTableView()

Big O(n) Linear time. This function is going through the SQL database through the link created in another function. It accesses the database and reads all of the info one by one from the table and stores it in the program through objects stored in a QList in QT.

void Graph::addEdges(const std::string &u, const std::string &v, int w)

Big O(n^2) polynomial time, this function initializes the adjacency matrix in the graph which is then used to calculate the distances and figure out if there is a path between the nodes or not.

QList<Team> dbManager::getAllTeams()

Big O(n^2) polynomial time. This function goes through the SQL database and reads all of the info from each team one by one and stores each team into a Team object and stores that team to the QList that holds all the teams.

double Graph::shortestDistance(std::string start, std::string end)

Big O(n^2) polynomial time. This function makes a priority queue and uses a while and for loop to go through the priority queue to get the shortest distance from the database.

void TripPlanner::initializeGraph()

Big O(n) Linear time. This function uses a map and vector and holds the edges in the vector. Then, the function creates a new graph and uses a for loop to add each edge to the graph one by one.