Faris Hijazi - Bobby Gaines - Myles Isas - Mohammed Taei

Big O

Login::on\_pushButton\_login\_clicked()

The Big Oh of this method is in O(1) because we already have the index and are just looking for a two elements to match. Since the check is happening in constant time, the index is already found when attempting to log into the admin panel or the user main panel. The input is taken instantaneously from the user and is checked against the conditions to ensure O(1) efficiency.

DbManager::addCity()

The Big Oh of this method is in O(N) due to it reading input loaded from the distancedata.csv file via the loadcityData function and inserting it into the database line by line increasing complexity and run time but is the most efficient method for insertion due to it being able to handle an unlimited size. The runtime is solely based on the size of the dataset being passed to it thus making it O(N).

DbManager::DbManager()

DbManager’s constructor is used to define the location of the database file and check whether if the database being accessed is valid or invalid (exists). The DbManager performs a path + “filename” to find the database but this does not impact the runtime due to it being predetermined at compile time. The runtime of this method is O(1) or constant time due to the database not being modified but instead just being accessed only.

Admin::ChangeFood()

ChangeFood allows the user to change or remove food items associated with cities within the database. It locates the foods by populating a List Widget via a QSqlQuery method and the Big Oh of this method is based on the runtime and size of the database. After the data is procured the elements still need to be populated and will be done so in O(N) due to it needing to be done on an element to element basis. The worst case scenario of the QSqlQuery along with listing the elements as well would be O(N2).

­TripPage:: tripPage()

The tripPage accumulates the cities selected by the user from the database displayed on either the MainWindow page or the Custom Trip planner page and stores the data into a linked list of pages individual trip stops or in the case of a basic trip individual displays the preset trip dependent on the user selection of amount of stops. This runs in O(N) due to the size and speed being entirely dependent on the size of the dataset selected along with the best case scenario being O(1) if the element list is limited to one element but no worse than O(N) due to the structure of the data being passed into it.