Angry Nerds
Amin
Meredith
Behrad
Awf
CS1D mw 2:30 PM
03/17/2021

College Tour BigO Analysis

1. Data Structures

   a) Vectors – This Vector list is the basis for both the tripPlanner and tripProgress classes.

- checkBoxvector: O(1)

Adding to the list is done by managing and updating checkbox for user selected colleges which lets addition to the list run in constant time.

2. AdminPanel
   a) AddColleges: Best Case O(n), Worst Case O(n^2)
      Each college must be added to the database, which gives an O(n) baseline run time right there. Adding each College can take anywhere from the best case O(1) to the worst case O(n) insertion time. If all Colleges take the worst case then this will run in O(n^2).

   b) addSouvenir: O(n)
      To add each souvenir in the database. This function runs in O(1) so the total run time is O(n).

   c) Remove a Souvenir: O(n)

      Removing the souvenir form the database can take up to O(n) time, but removing it from the list requires a search of the database which will guarantee a run time of O(n).

   d) Change a Souvenir: O(n + m), n = # of colleges, m = # of souvenirs
      Changing a Souvenir might require a resort of both the college list and the souvenir lists. Each of these would require a search of their respective lists.

3. Trip Planner

   a) Adding a Single Starting College to the Trip: Best Case O(1), Worst Case O(n)

Accessing the starting college is done with a query lookup which has an expected run time of O(1) but which can take up to O(n).

b) Adding closest closest colleges starting at Arizona state university to the Trip: Best Case O(m), Worst Case O(n)
n = # of colleges in total
m = # of colleges planned to visit during the trip

Accessing the starting college is done with a query lookup which has an expected run time of O(1) and worst case time of O(n). To remove the colleges we must iterate the list of planned colleges for the trip, which takes O(m) time.

c) Buying a Souvenir from a given college: Case O(n)
n = # of souvenirs infall the college

A list of souvenirs must be iterated which gives the O(n) base time. Purchases are stored in a list but insertion to the list takes O(1) time because new orders are inserted to the back.

d) Calculating Trip Route: O(n * m^2)
n = # of colleges in total
m = # of colleges planned to visit during the trip.

The limiting factor on run time for this operation is the planAlgorithm function in the graph, which runs in O(n*m^2) time.