Bryan Indelicato 10/22/20
Professor Chen CSE248

In my hiking project I will be evaluating which data structures are most optimal to store user information, hiking trail data, as well as a log for hiking history. Each section of this project will have the opportunity to use one or multiple data structures. I'm already able to determine what data structures are invaluable to certain sections of the project. For the user profiles I have eliminated the structures: Array, Hashmap/ Hashsets, Queues, and Stacks. These decisions were obvious based upon the needs of the users profile, I cannot use anything Array based because the size of the profiles will always be changing depending on how many people sign up. So to create an array with a pre determined size would be very constraining when the array gets possibly full. This also applies to the hiking trail data because there is no predetermined size of how many trails there will be and the number will keep growing and then fill up the array.

For the project I think the most optimal data structure for the user profiles will be a hash table as it will allow very fast insertion, search, and deletion at O(1) time but worst case O(n) time. Im going to compare this to a linked list data structure as I think thats another option for the storing the user profiles, as it has very fast insertion and deletion(O(1) time) but slightly slower searching and accessing time(O(n) time). For storing the trail data I think the best data structure would be using a TreeMap as it allows for sorting the data set automatically as well as the searching is fast a O(logn). I will be comparing the TreeMap to a Linked List as it's a possible alternative to the TreeMap. For storing the hiking history in user profiles I decided that I will only use a linked list as its the only logical data structure as it can keep order of the trail history as well as has no requirements in size, and does not need to have fast indexing or searching.

Bryan Indelicato 10/22/20
Professor Chen CSE248

For testing my hypothesis I have created the data structures to be directly compared with the same data set and values. For the user profiles I will be test how long it takes to create the HashTable and LinkedList by inserting varying sample size data from 0 to 67000 unique user profiles. Each data point will contain the average of 5 tests for an accurate average of time to insert data to the data structures. Every time the data set is loaded into the HashTable and LinkedList that data will be randomized at the beginning of the test. The second test will be comparing the user profiles data structure search time complexity. The test will search for one unique user profile randomly indexed in the data structure for fair and accurate results. For the trail data I will be conducting the same exact style of test as the user profiles but instead using unique trail data set.

Based upon my results with my fictional data set I have determined that the user profiles data is optimally stored on a hash table as it provided the fastest search time for accessing user data, this is good for having the fastest log in management. I have also decided that the best data structure for the trail data is the tree map as it provides no restriction on size requirements as well as its has constantly outperformed the linked list data structure. The only issue that I saw with the data was the time it took for some of the test had outliers from my computer performing background operations. I think in the future in order to get a more refined data I will need to figure out a method to time the complexity of the data structures without interference from the CPU being used by other programs.