

Snowday

The Snowday project is the idea of storing ski resorts and the city of Boulder and Denver in a weighted graph to create an interactive program in which the user can view the graph, the distances between locations, add resorts and connections as well as finding the shortest path for their trip and explore the runs on each resort. In order to do so, we decided to store 100 runs from each resort on a text file and as the user asks for a specific resort, the runs are read from the file and stored in a linked list or a hash table. After stored, the user can ask to view all the runs from a specific resort or to find out more details about a specific run, among other functions with the graph. The objective of the project was to create this interactive program using data structures learned in class and finally compare the time it takes for the program to find a run in a linked list with the time taken to find a run in a hash table. In order to implement this project we used the following data structures for the following purposes:

Graph

- Used to create a linked map of the ski resorts near Boulder and Denver
- Weighted graph with the distances between the ski resorts, Boulder and Denver
- Each vertex stores the name of the location
 - Vail vertex is pointing to a linked list with contains in each node the name and type of a run/lift
 - Breckenridge is linked to a hash table containing the name of the runs

Linked List

- The Vail vertex points to a linked list to store runs of the resort.
- The nodes in the list contain the names of the runs and the type (lift, green run, blue run, black run, double back run, terrain park).
- The list contains 100 of Vail's runs with the objective of comparing time complexity with hash tables.

Hash Tables

- The Breckenridge vertex is linked with a hash table that stores the runs and lifts from the resort
- Each position in the hash table contains the name of a run and the type is later on found if a user asks for a specific run
- The table contains 100 of Breckenridge's runs with the objective of comparing time complexity with linked lists

All the data used in the project was taken from the official Breckenridge and Vail website to create the text files in which the runs are stored. Each resort has its own separate text file and the runs are stores in the format: Run Name,Type. The file is read and the runs from Vail and from Breckenridge are stored in a linked list and a hash table respectively. Each node in the linked list contains the name and type of the runs in different variables and the hash table contains the names of the runs, to print all the runs perfectly. The type of each run in the hash table is taken from the file upon request of the user. The goal of the project was to compare the efficiency of hash tables when compared to linked lists while creating something fun.

After testing the program we arrived at the conclusion that hash tables are more effective in locating the runs most of the time. Even though the method used to solve collisions was linear probing, hash tables proved to be more effective in most cases.

We met through the weeks to finish the project and worked together on every aspect of it. The workload was perfectly distributed and every member of the group got to work with every function in the program.

Visual representation of the program is below:

