MAC 286 Data Structures Fall 2020 ()

Discussion 1: Part 2

Instructor: Prof. Oehrlein

Part 2
You will read two responses to the prompt (reprinted below) from your classmates. Write a short response (you can just email me!) to the following questions.

- What do the two responses agree on? disagree on?
- Is there anything you agree with in your peers' comments? Disagree with?
- What would your choice of a data structure and implementation in this situation be now, and is that different from before? Why/why not?

Below is a reminder of the prompt, followed by the set of representative responses.

A reminder of the prompt
Exits from interstates in the US often have names and numbers. The name is usually that of the street, town, or city that the exit goes to. The number is generally a distance in miles from a basepoint, like a state border. Think about a data structure for a sequence of some interstate exits, each with a String name and an int number. We will want to be able to do the following things with the structure:

- Given the name of an exit, return the corresponding distance from the basepoint.
- Given an exit number, return the exit name.
- Given a distance from the basepoint, find the nearest exit.

What data type would you use for this problem? Choose one and explain your choice based on what we want to be able to do with the structure.

Responses

1. This problem can be solved by different approaches. However, I choose binary search trees as the way I would use for this problem. Binary trees use nodes which will allow me to store different data types (String for the exit name and int for the number). Also inserting and searching for a node in a binary tree is easier and more efficient than using other data structures (the time complexity for the insert and search operations is O(log (n)). Moreover, since for this problem we need to find the name of the exit given the distance from the border, and the name of the exit given the number, we can use two find methods one with a String parameter and the other with an int parameter. The find(int) method will compare the value given with every int field in the tree (moving to the left if the value given Is less or to the right if it is greater) until the node holding that integer value is found, while the find(String) method will compare the name given with every exit name in the tree until the node holding that name is found. Each method will then return the corresponding value to the name or number of the exit. Furthermore, to find the nearest exit given a distance from the border we can start by sorting and inserting the exits in the tree according to their distance from the border instead of the exits' names. In that way we can use a measurement method to find the nearest exit by comparing the given distance to the int field in every node starting from the root node until we find the nearest exit from that distance. By using binary trees, the process of finding the data wanted will be easier and faster than using other data structures like linked lists or arrays which are less efficient. Therefore. Binary Trees is my choice for this problem.
2. The best data structure to use for this would be ordered arrays. With ordered arrays, we can use binary search to efficiently find a particular exit. Because we would not expect to add or remove highway exits very frequently, the primary downside of using arrays is somewhat mitigated.