Eduardo Robles

CS-163

Efficiency Review Program #2 Write-Up

For this week’s program we designed a trip planner program that would keep track of all the places a user visits and feedback about the stay. In order to store the information, we used used stacks and queues. For the implementation of the queue we used a circular linked data structure. The stacks used a combination of linear linked list and arrays. I think these data structures performed well and were a good option for this type of task. The linear link list aspect in both meant that we didn’t have to worry about running out of memory or shifting data. We could keep on adding more trips because the memory is being dynamically allocated. The linked list of arrays was super efficient and quick. It lets us not worry about memory and still gives us that direct access.

For this program instead of linked list we could have used dynamic arrays for the data structure. Arrays are faster and have direct access to all its elements. This would make it really easy to access and add trips to the stacks and queues. The only problem with this implementation is that we have to the size of the array beforehand. This would limit to the amount of trips we could have and would end up requiring to shift data around which would slow everything down. If the number of trips is known or set beforehand, then this would have been a perfect data structure.

The use of linear linked list of arrays was the most efficient part of the design. It allowed for the most flexibility but still lets us grow as we needed. All functions passed everything by reference so no extra copies were created. The same stop object was used for both the stack and the queue.

I couldn’t figure out how to edit existing items on the stack without creating more stop objects. I wasted more memory than I probably needed implementing that feature. If I had more time I would figure out how to fix that problem and maybe experiment with arrays and maybe try out array of linked list instead of linked list of arrays. Overall I think core features were implemented correctly and used the data structures to their advantage.