Module 4: Critical Thinking

List Based Stack

Nolan Byrnes

CSC506 – Design and Analysis of Algorithms

Colorado State University – Global Campus

Professor Isaac Gang

October 7th, 2022

Module 4: Critical Thinking

Geeksforgeeks defines stacks and queues both as a “linear data structure in which elements can be inserted and deleted only from one side of the list” (Geeksforgeeks, 2022, para. 1). The difference is that stacks follow the last in first out principal (LIFO), whereas queues follow the first in first out principal (FIFO). Two linear data structures that can be used for a stack and a queue are a Python list, and a doubly linked list, so I tested and compared the performance of the Python list-based stack and queue, to the stack and queue based off a doubly linked list by timing how long it took to insert and remove 500,000 random numbers into the respective stacks and queues. I had it to where each test inserted and removed the same list of 500,000 random numbers and used milliseconds (ms) for the time measurement. Observe the results in figure 1 below.

Figure 1.

Performance test results for python list and doubly linked list based stacks and queues

Graphical user interface, text

Description automatically generated

Note. The results represent the amount of time in milliseconds it takes to include and remove 500,000 random numbers from Stacks and Queues with a Python List and Doubly linked lists.

Python’s documentation on lists mentions that it is “possible to use a list as a queue, where the first element added is the first element retrieved (‘first-in, first-out’); however, lists are not efficient for this purpose” (Python, 2022). The reasoning behind the inefficiency of using a Python list for a queue is that for each time that an element is popped from the 0th index, all other elements need to shift by one. This explains why it took a significantly longer time for the list-based queue to remove all 500,000 elements is because since the queue was removing the first index of the list, each time that it was popped, every other index had to be shifted -1 index. If you think about a line at a cafeteria, each time that the person in the front of the line is served, and leaves the queue, it takes a lot of effort for everyone to move one position forward. It took just about the same about of time for the list-based stack to insert and remove the elements, because each time that an element was popped from the stack list, the last index was removed and none of the element’s shift.

Lysecky defines a doubly linked list as “a data structure for implementing a list ADT, where each node has data, a pointer to the next node, and a pointer to the previous node” (Lysecky, 2019, para. 1). Since each node has a pointer to its next node, as well as its previous node, it does not take much time to add or remove elements to the front of the list, as none of the nodes will have to shift, unlike with the Python list-based queue. Geeksforgeeks explains that the disadvantages of having a doubly linked list are that “it uses extra memory when compared to the array” and “since elements in memory are stored randomly, therefore the elements are accessed sequentially no direct access is allowed” (Geeksforgeeks, 2022a, para 3).

For the doubly linked list implementation of the stack and queue, it took about the same time to include and pop the items. Comparing the doubly linked list and stack to the list-based stack and queue, it took longer for all the elements to be included in the doubly linked list and queue since for each number that was to be included, it had to create an instance of the Node class, and for each node, the next or previous node of each node had to be assigned. Whether the program was adding nodes to the head or the tail of the doubly linked list, it would take the same amount of time because it is assigning the new node as either the head or the tail. The same goes with popping the doubly linked list or queue, as it would be removing either the head or the tail, and none of the elements in between would have to shift anywhere.

**REFERENCES**

GeeksforGeeks. (2022, July 14). Difference between stack and queue data structures. https://www.geeksforgeeks.org/difference-between-stack-and-queue-data-structures/

Geeksforgeeks. (2022a, July 26). Advantages, disadvantages, and uses of doubly linked list. https://www.geeksforgeeks.org/advantages-disadvantages-and-uses-of-doubly-linked-list/

Python. (n.d.). 5. Data structures — Python 3.10.6 documentation. 3.10.6 Documentation. <https://docs.python.org/3/tutorial/datastructures.html>

Lysecky, R., & Vahid, F. (2019). 4.7 Doubly-linked lists. In Design and Analysis of Algorithms. zyBook.