

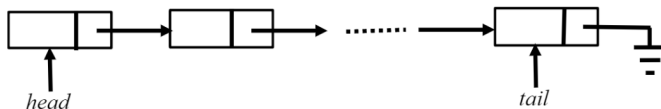
EE4013 - C and Data Structures

Tejaswini A V
EE18BTECH11047

October 13, 2021

Data structure - Queue

Problem: A queue is implemented using a non-circular singly linked list. The queue has a head pointer and a tail pointer, as shown in the figure. Let n denote the number of nodes in the queue. Let *enqueue* be implemented by inserting a new node at the head, and *dequeue* be implemented by deletion of a node from the tail.



What is the time complexity of the most time-efficient implementation of *enqueue* and *dequeue*, respectively, for this data structure?

Solution - Enqueue

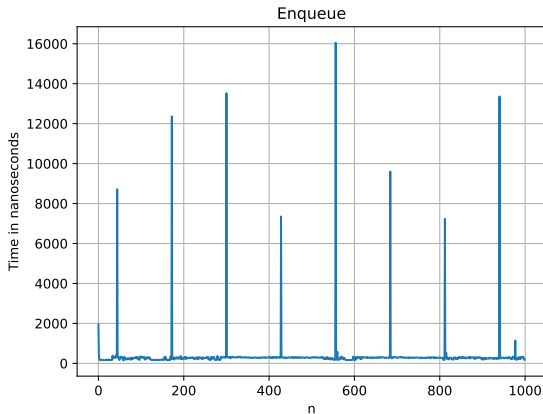
Enqueue: The enqueue operation involves two operations:

- Create a new node with the value to be inserted and point the next pointer of this node to the head of the linked list.
- Make the new node as the head of the linked list.

Creating a new node and changing two pointers can be performed in constant time. Therefore, time complexity of *enqueue* operation is $\Theta(1)$.

Enqueue - Time complexity

The following plot shows the experimental time complexities of Enqueue operation for various values of n .



Deque

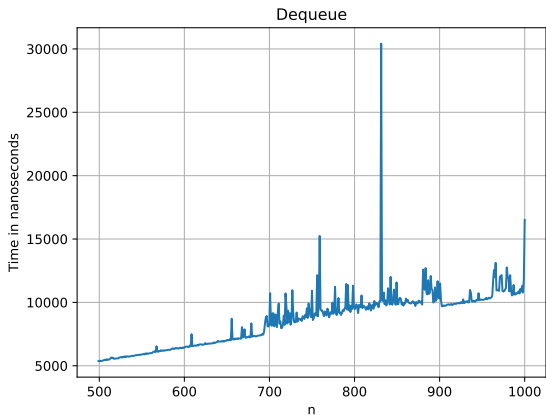
Deque: The dequeue operation involves the following steps:

- Delete the tail node.
- Traverse the linked list to find the address of the second last node and make the next pointer NULL. Update the tail pointer.

Since we are traversing the whole linked list, the time complexity of *dequeue* operation is $\Theta(n)$ where n is the number of nodes in the linked list.

Deque - Time complexity

The following plot shows the experimental time complexities of Dequeue operation for various values of n .



Time complexity

Range of time taken:

Enqueue - $[0.156, 0.412] \mu \text{ sec}$

Dequeue - $[5.356, 12.756] \mu \text{ sec}$

Approximate equation of the time taken by dequeue is

$$n_1 = 500; t_1 = 5.356 \mu \text{sec} \quad (1)$$

$$n_2 = 1000; t_2 = 12.756 \mu \text{sec} \quad (2)$$

$$\text{slope}(m) = \frac{t_2 - t_1}{n_2 - n_1} = 0.0148 \quad (3)$$

$$\text{Intercept}(c) = t_1 - m * n_1 = 5.356 - 0.0148 * 500 = -2.044 \quad (4)$$

$$\implies t = m * n + c \quad (5)$$

$$\implies t = 0.0148 * n - 2.044 \quad (6)$$

where n is the number of nodes in the linked list and t is the corresponding time taken for dequeue operation in micro seconds($\mu \text{ secs}$).

Conclusion

Therefore, the time complexity of the most time efficient implementation of *enqueue* and *dequeue* respectively are $\Theta(1)$ and $\Theta(n)$.