

## 1) Big O Time Complexity Analysis

\*addPatient

This function adds a new patient to the end of the list. If the list is not empty all items are checked one by one.

n= number of patients on the list

Time complexity is  $O(n)$ . Because entire list is traversed to reach the last element.

\*removePatient

This function scan the list from the first element to find the patient to be removed.

All items are checked one by one.

Time complexity is  $O(n)$ . Because all elements are checked one by one.

\*findPatient(int id)

This function search the linked list to find a patient with the given id information.

Time complexity is  $O(n)$ . Because if the patient is first element of the list, time complexity is  $O(1)$ . If the patient is last element of the list, all elements are checked one by one.

## 2) Compare the performance of Linked List vs. ArrayList

\*addPatient

For linked list: Time complexity is  $O(n)$ . To reach the last of the linked list, the list is checked one by one.

For Array list: Time complexity is  $O(1)$ . The element is appended to the end.

\*removePatient

For linked list: Time complexity is  $O(n)$ . We need to find the element for deleted.

For Array list: Time complexity is  $O(n)$ . Scrolling occurs after the element is deleted.

\*findPatient(int id)

For linked list: Time complexity is  $O(n)$ . Linear search is used.

For Array list: Time complexity is  $O(n)$ . Linear search is used.