

Section D: Asymptotic Computational Complexity

Option 2: Time Complexity

Please report on the following:

- The worst-case time complexity of the “Collection.add” method.
- At least two alternative designs that would improve the worst-case time complexity without using arrays or any other predefined collection type.
- The worst-case time complexity of the alternative designs.

Answers

- The worst-case time complexity of the “Collection.add” method.

The worst-case scenario is to traverse through the collection(linked list) until we reach the tail. The tail is when the next value is null, so we change the value of the new collections to point to the tail pointer, and the new collection will be null.

The time complexity for this operation is linear $O(n)$, since, for each addition, we are traversing to the end of the tail.

- At least two alternative designs that would improve the worst-case time complexity without using arrays or any other predefined collection type.

Alternative 1:

One can improve on the existing linked list(collection) with a double linked list. This provides each collection with two pointers; a pointer to a previous node and a pointer to the next node. This helps to add collection both ways(head and tail), with a constant time of $O(1)$.

- The worst-case time complexity of the alternative designs.

The worst-case time complexity is $O(1)$