



AlgoTutor Web Application

Integrated Learning Platform for Data Structures & Algorithms

Student: Tan Zong Wei

Supervisor: Professor Owen Newton Fernando

linked list 1

What is a linked list in the context of data structures?

- A collection of elements arranged in a linear order
- A two-dimensional array
- A hierarchical data structure
- A collection of key-value pairs

Which of the following is a disadvantage of using a linked list compared to an array?

- Faster access to elements
- Fixed size
- Slower insertion and deletion operations
- Inability to store different data types

In a singly linked list, each node contains:

- Data and the address of the previous node
- Data and the address of the next node
- Data and both the address of the previous and next nodes
- Data only

```
1 //////////////////////////////////////////////////////////////////
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 //////////////////////////////////////////////////////////////////
7
8 typedef struct _listnode
9 {
10     int item;
11     struct _listnode *next;
12 } Listnode; // You should not change the definition of Listnode
13
14 typedef struct _linkedlist
15 {
16     int size;
17     Listnode *head;
18 } LinkedList; // You should not change the definition of LinkedList
19
20
21 ////////////////////////////////////////////////////////////////// function prototypes //////////////////////////////////////////////////////////////////
22
23 // You should not change the prototype of this function
24 void moveElementsToBack(LinkedList *ll);
25
26 void printList(LinkedList *ll);
27 void removeAllItems(LinkedList *ll);
28 Listnode * findNode(LinkedList *ll, int index);
29 int insertNode(LinkedList *ll, int index, int value);
30 int removeNode(LinkedList *ll, int index);
31
32 ////////////////////////////////////////////////////////////////// main() //////////////////////////////////////////////////////////////////
```

Input Output Analysis

Analyze Code Time Complexity Space Complexity

```
1 //////////////////////////////////////////////////////////////////
2
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 //////////////////////////////////////////////////////////////////
7
8 typedef struct _listnode
9 {
10     int item;
11     struct _listnode *next;
12 } Listnode; // You should not change the definition of Listnode
13
14 typedef struct _linkedlist
15 {
16     int size;
17     Listnode *head;
18 } LinkedList; // You should not change the definition of LinkedList
19
20
21 ////////////////////////////////////////////////////////////////// function prototypes //////////////////////////////////////////////////////////////////
22
23 // You should not change the prototype of this function
24 void moveElementsToBack(LinkedList *ll);
25
26 void printList(LinkedList *ll);
27 void removeAllItems(LinkedList *ll);
28 Listnode * findNode(LinkedList *ll, int index);
29 int insertNode(LinkedList *ll, int index, int value);
30 int removeNode(LinkedList *ll, int index);
31
32 ////////////////////////////////////////////////////////////////// main() //////////////////////////////////////////////////////////////////
```

Linked List Vizualiser

Add Node by Index Enter a index

Delete Node by index Enter a index

Delete Node by Value Enter node value

Project Objectives:

The objective of this project is to develop an integrated web application with an in-built AI support using ChatGPT, to assist students in the learning process for Data structures and Algorithms. The webpage will be an integrated platform where students will be able to learn theoretical concepts, practice coding-based problems, as well as view interactive visualizations. The application is developed with the FReMP Stack, consisting of Python Flask, MongoDB, and ReactJS, and custom visualizations are developed using CanvasJS.

Features:

- **Interactive and original data structures visualizations**
- **Online C code compilation** and problem-based learning
- **MCQ Quizzes** for learning of theory
- **Tailored Code Analysis** and personalized guidance
- **Retrieval Augmented Prompt Engineering**

