Linked List - ADT

Author

Keywords:

Node: an element of the List

Pointer: a variable that stores the address of another node it points to.

NULL Pointer: a pointer that doesn't point to anything.

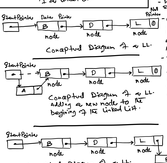
Start Pointer: a variable that stores the address of the first element of an linked list.

Rules

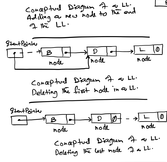
- All elements, nodes data, should be inserted in order.
- Start is indexed 1.
- Start is constant.
- use count go back, only can be found with pointer.
- Number of nodes is unknown.
- Dynamic delete & insert.

Functions

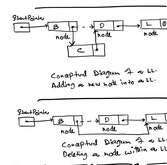
- Insertion a node: It should be in order. It can be made available at the start, the middle or the end.
- Delete a node: It should be found and then deleted.
- Traversal: Print a node, It traversal/Added to the LL, it will be taken from the start of the free list.



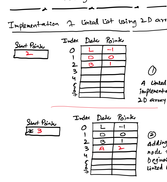
Conceptual Diagram of a LL.



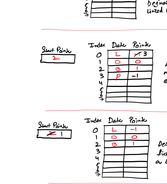
Conceptual Diagram of a LL. Adding a new node to the beginning of the Linked List.



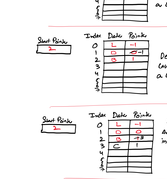
Conceptual Diagram of a LL. Adding a new node to the end of a LL.



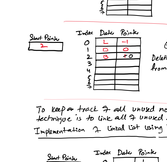
Conceptual Diagram of a LL. Deleting the first node in a LL.



Conceptual Diagram of a LL. Deleting the last node of a LL.

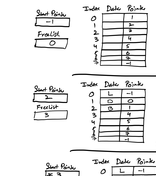


Conceptual Diagram of a LL. Adding a new node into a LL.

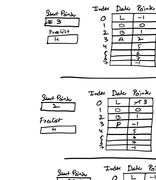


Conceptual Diagram of a LL. Deleting a node within a LL.

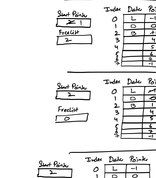
Implementation of Linked List using 2-D array:



① A linked list implementation using 2D array.

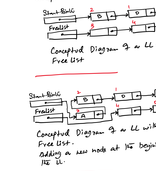


② Adding a new node to the beginning of a linked list.

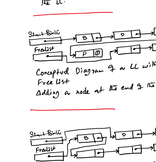


③ Adding a new node to the end of a LL.

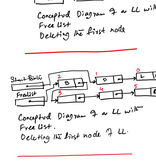
④ Deleting the first node in a LL.



⑤ Deleting the last node of a LL.



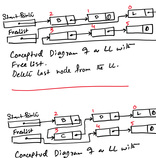
⑥ Adding an item into a LL.



⑦ Deleting an item from a LL.

To keep track of all unused nodes as decreases, so we give all of unused nodes, free list.

Implementation of linked list using 2-D array:



Setting up the linked list

A linked list implementation using 2D array.

① A linked list implementation using 2D array.

② Adding a new node to the beginning of a linked list.

③ Adding a new node to the end of a LL.

④ Deleting the first node in a LL.

⑤ Deleting the last node of a LL.

⑥ Adding an item into a LL.

⑦ Deleting an item from a LL.

Conceptual linked list usage with free list:

Conceptual Diagram of a LL with Free List

Conceptual Diagram of a LL with Free List. Adding a new node at the beginning the LL.

①

Conceptual Diagram of a LL with Free List. Adding a node at the end of the LL.

②

Conceptual Diagram of a LL with Free List. Deleting the first node

③

Conceptual Diagram of a LL with Free List. Deleting the last node of a LL.

④

Conceptual Diagram of a LL with Free List. Adding an item to the LL.

⑤

Conceptual Diagram of a LL with Free List. Deleting an item from LL.

⑥