

Chapter 2: Lists: Linked Lists, Stacks, Queues

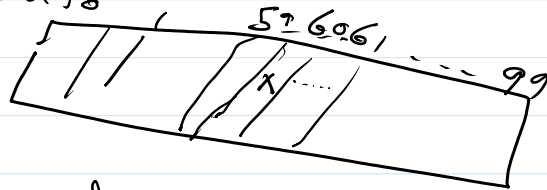
Wednesday, March 10, 2021 11:18 AM

* if we have group of elements, we have used arrays.

* Arrays: static data structure, its size is fixed.

x at index 60.

$$A[60] = x$$



insertion and deletion of elements at certain position is time consuming

Linked list: is a dynamic data structure, consists a group of elements which together represent a sequence.



* Basic operation

<u>Operation</u>	<u>Array</u>	<u>Linked List</u>
creation	$O(1)$	$O(1)$ —
Insert	$O(n)$	$O(1)$ ↑
Delete	$O(n)$	$O(1)$ ↑
Access of an element	$O(1)$	$O(n)$ ↓
Point the list	$O(n)$	$O(n)$ —
Search	$O(n)$	$O(n)$ —
make Null	$O(n)$	$O(n)$ —
strategy	Static	Dynamic ↗

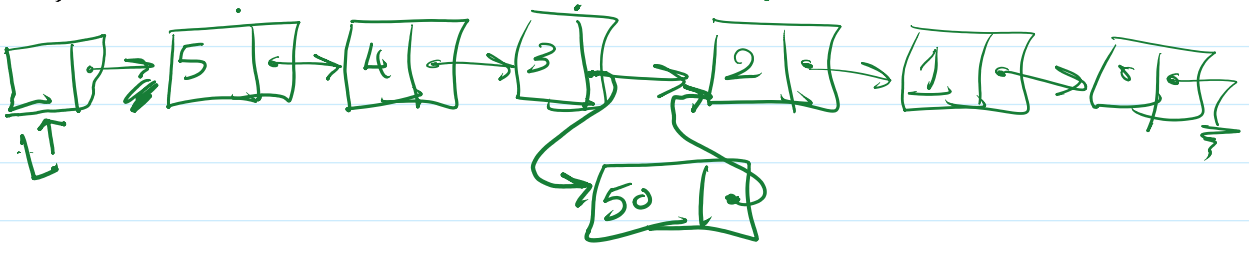
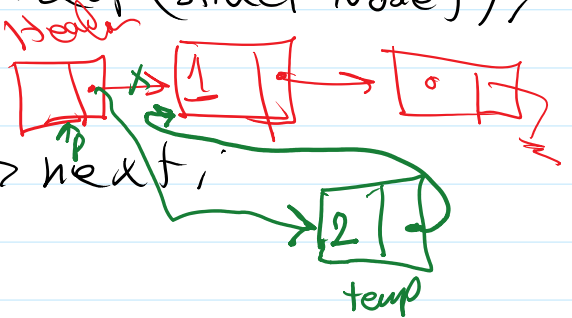
* Adv. L.L.:-

- ① Easy to store data of different sizes.
- ② Flexible and dynamic data structure.
- ③ Insertion and deletion of elements do not require shifting of elements.


```

void insert(List L, Position p, int x) {
    Position temp;
    temp = (Position) malloc(sizeof(struct Node));
    if(temp != NULL) {
        temp->element = x;
        temp->next = p->next;
        p->next = temp;
    }
    else
        printf("sorry: memory is full");
}

```



* لإضافة العنصر الجديد في آخر القائمة
 $O(n)$
 $O(1)$