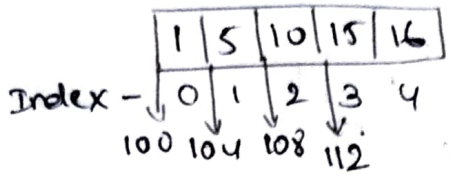
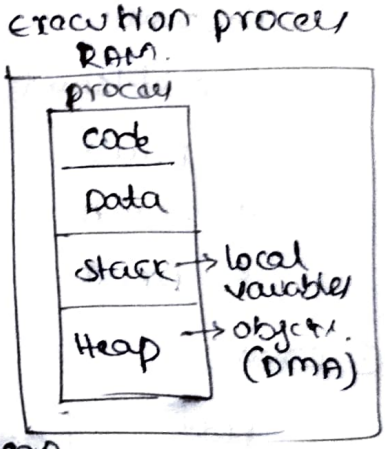
Data structures 16-Sep-2025

Array - stores linear data type [Homogenous] in continuous memory location.

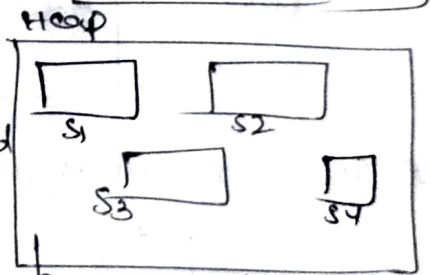



Int - 4 bytes / 2 bytes
Depends on OS.

* Normal array in java support primitive Datatype
* Singly Linked List
* Array has disadvantage in order to static declaration
* Linked list has Dynamic memory allocation (DMA)




* For deallocation of this objects in C/C++ we use free()
* Garbage collector - can be configured now inorder to delete the used objects.
* Garbage collector refers counter. If counter become 0 so it will clear the process.

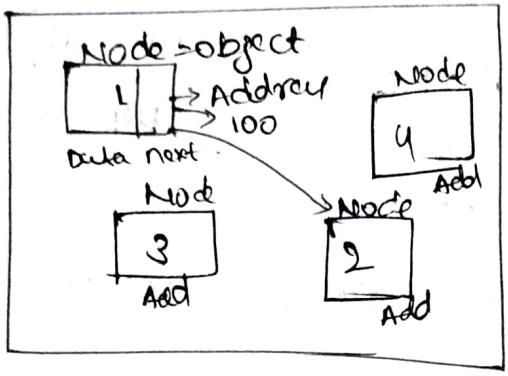



Dynamic allocation - If it is full it refers to out of memory.
→ we leave incase of deallocate the process this is called memory leak

* But some time Garbage collector has also leave objects becoz managed code it has reference ID of object. then this refers also memory leak.

* In each Heap

* In Heap, each node treated as object and also it has address



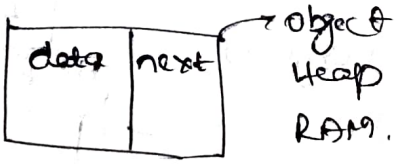

* singly linked list is one way. but in Doubly linked list has two way.

Basic Operation

1> Create
2> Add / Insert
3> Delete
4> Search
5> print

① Create

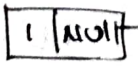
Node -




② Insert → At Beginning
Any position.
At end

At Beginning

```
Node new_node = new Node(1);
```




```
new_node.next = head;
head = new_node;
If head=empty;
```

At end.

```
current_node = head
while (CURRENTNode != Null)
{
    currentNode = current
    Node.next
}
Currentnode_next =
```

## Time complexity

i) Identify Basic operations.

ii) count the Basic operations.

iii) express the count as a function.

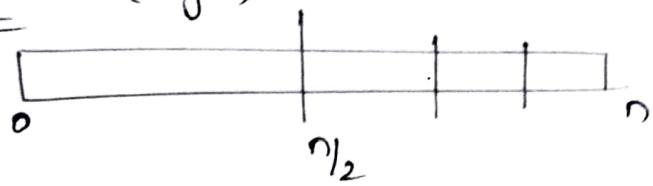
iv) simplify the function

v) use Big O Notation.

### const O(1)

It include only constant operation - directly.

### Logarithmic O(log n)




It Divides

### Linear O(n) – It includes all search.

### Quadratic $O(n^2)$.

### cubic ($O(n^3)$)

### Exponential $O(2^n)$.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10. |
|---|---|---|---|---|---|---|---|---|---|

subjects.

### Factorial O(n!)