23-Dec-2021

23 December 2021 20:50

HashSet:- 1, 2, 3, 4,

get => O(1)

add = O(1)

remove = O(1)

1 2 3 4 5 6

Print common Elements:-

a = [1, 2, 3, 4, 5]

b = [3, 4, 5, 6, 7, 8]

n^2, n, (m×n),

with HashSet

O(n), O(1)

n, n^2, n log n ✓ O(1)

Space

TC

HashSet

a = [4, 1, 3, 5, 2,]

b = [6, 7, 1, 3, 2]

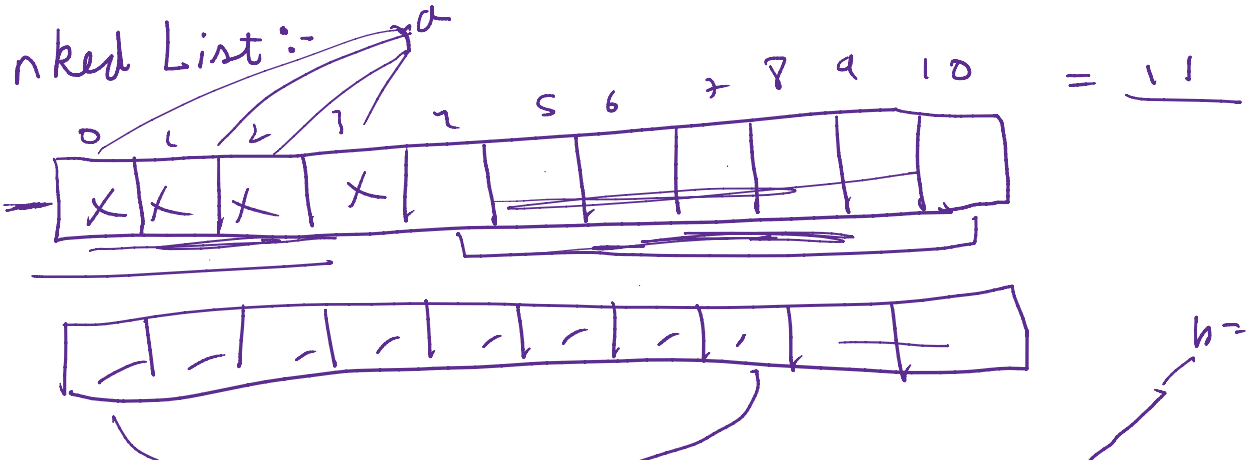
4
1
3
5
2

O(n), n^2, (m+n), 2n

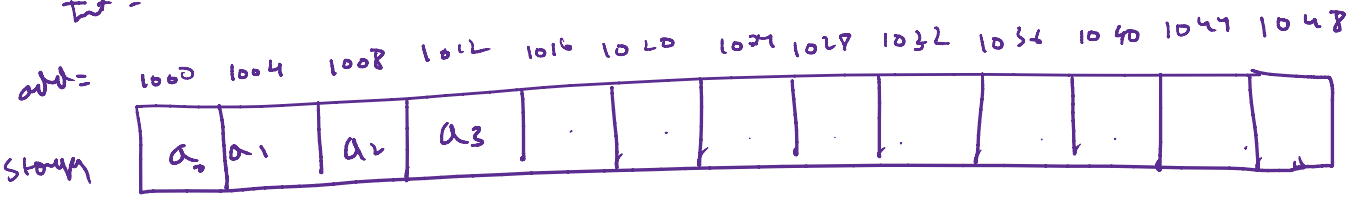
O(n)

[1, 3, 2]

Linked List:-



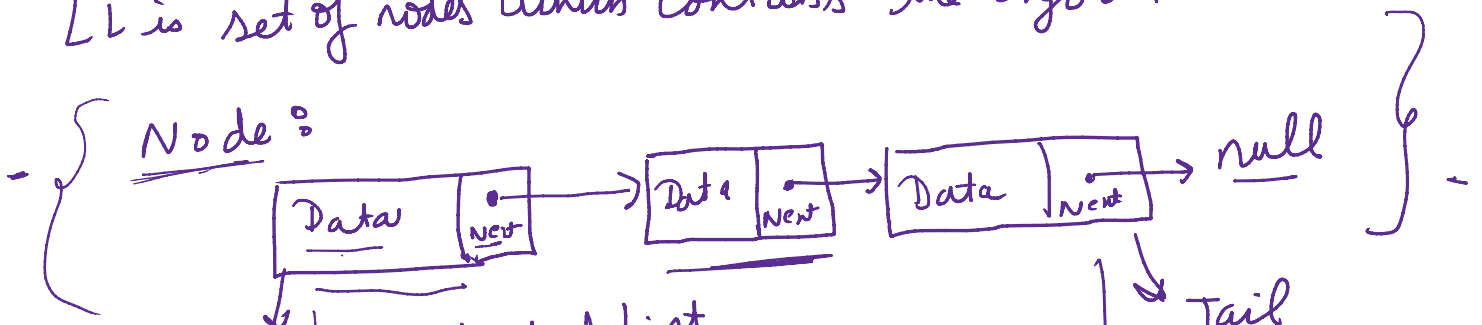
Int = 4 byte

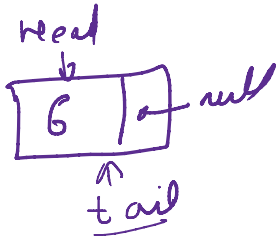


a = new [4]

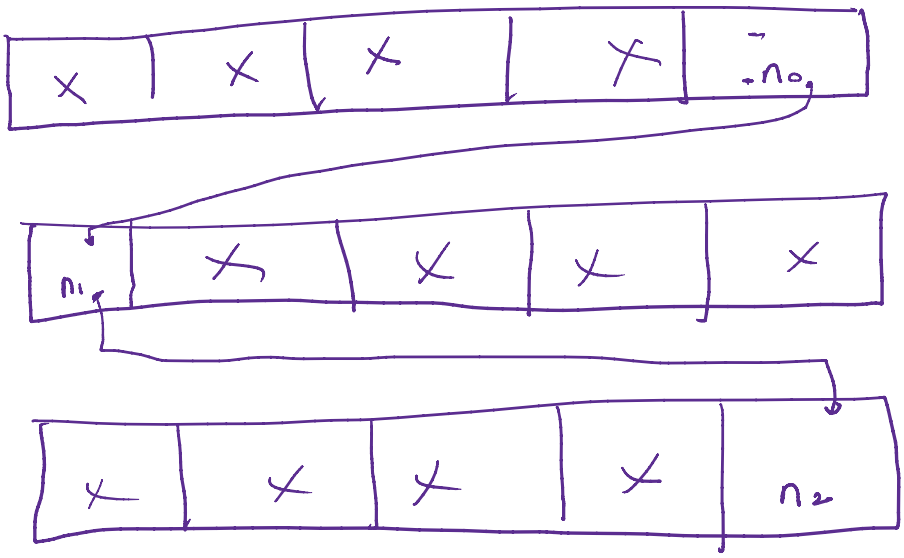
b = new [10]

LL is set of nodes which contains the information.





L.L



Linked List {

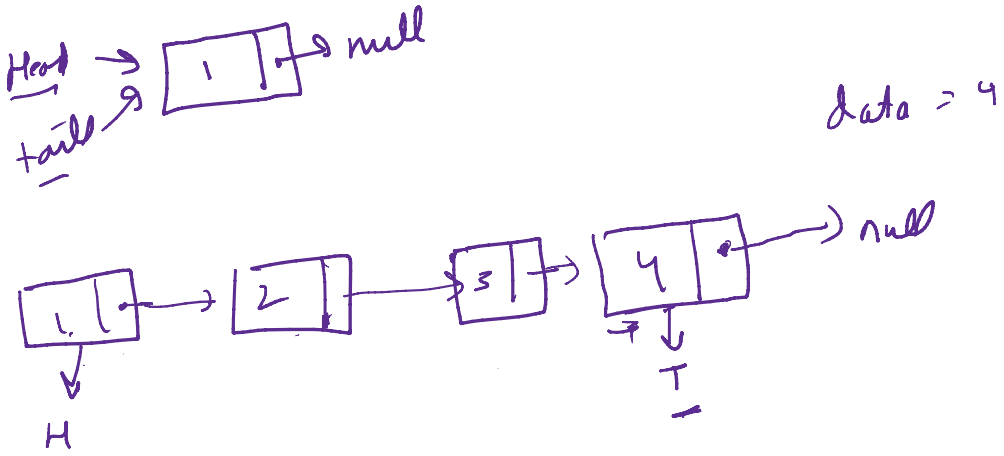
~~class~~ Class Node {

addLast (data) { O(1),

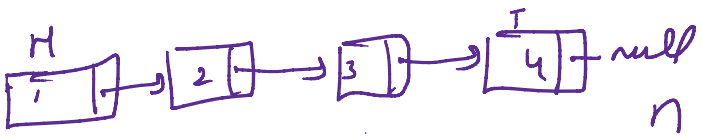
```
class Node {
    int data;
    Node next
}

Node tail;
Node head;
int size;
```

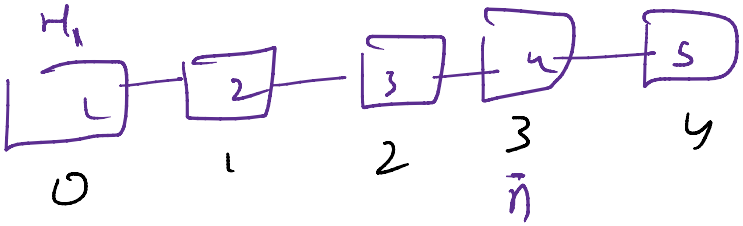
```
addLast(data) { O(1)
node = new Node(data)
if(tail == null) {
    tail = head = node;
} else {
tail.next = node;
tail = node
}
```




```
PrintLL() { O(n)
Node node = head;
while(node != null) {
print(node.data);
node = node.next
}
}
```




```
getAt(idx) { O(n)
i = 0;
```

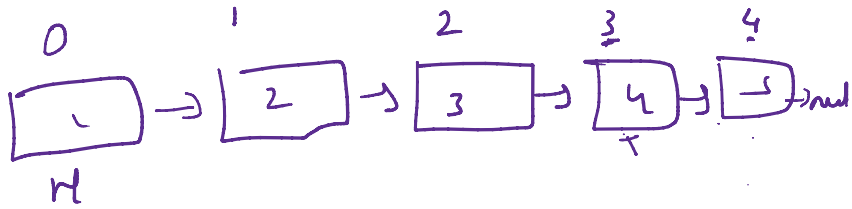


0 1 2 n

```
i=0 ;
node = head;
while (i < idx) {
    i++;
    Node = node.next;
}
return node
}
```

idx = 3

```
removeLast () { O(n)
node = getAt(size - 2);
Node.next = null
tail = node
size--;
}
```



size = 5