## Lecture

* Why array over arraylist
  + Size doesn’t change which can save memory
  + Slight efficiency boost
  + Primitives
  + Gaps in data
* LinkedList
  + Chain of nodes
    - Node<T> {  
       T data;

Node<t> next;

}

* Add Front
  + Head becomes the new node added
  + New head points to old head
  + NewNode = new Node(data);

newNode.next = head;

head = newNode

size++;

* + O(1)
* Add Back
  + Edge case: head = null
  + curr = head

while curr.next != null:

curr = curr.next

newNode = new Node(data)

curr.next = newNode

size++;

* + O(n)
    - Traverse n Nodes
* Add To Index (data, index)
  + Check index in bounds
  + Edge case: head is null
  + curr = head

counter = 0

while counter < index:

curr = curr.next

newNode = new Node(data)

newNode.next = curr.next

curr.next = newNode

* + O(n)