**DATA STRUCTURE**

Q: Dynamic arrays?

A: \_Usually an array cannot grow, but most languages have “arrays” that can grow. It’s called a dynamic array

A dynamic array wraps a real array and allows it to grow. How? Once the array fills up, a new, larger, memory location is found and the original is copied to the new space

+) Array’s size doubled but not incremented b/c the act of copying is expensive, so might as well

Q: But what about array.push?

A: So array in JS isn’t real “arrays.” They’re hashes that just happen to have integer values for the key

+) So array.push modifies the array dynamically

\_Arra is stored as one contiguous block of memory

**LINKEDIN LIST:**

Q: What’s that?

A: So 1 node consists of the value, and the memory address/ position of the next node.

The last node in the linkedin list will have position of “null.”

To traverse, it’s O(n)

Insertion is also O(n), but we don’t have to shift like we do with arrays. We just need to route the second block of the node of the value after we want to insert in to the new element, and make its address pointing to the next node in the linked list

Q: Advantage of linked list?

A: No need to block out a specific amount of memory like we do with arrays

Q: how to implement linked list?

A:

if empty, return NOT\_FOUND

index = 0, temp = head

while (temp.item != v)

index++, temp = temp.next

if temp == null

return NOT\_FOUND

return index

Q: Stacks & Queue?

A: Stack –LIFO (like a stack of plates)

+) Use linked list: from top to bottom

Queue – FIFO (like a queue to a movie theater)

+) Use a linked list (from left to right)