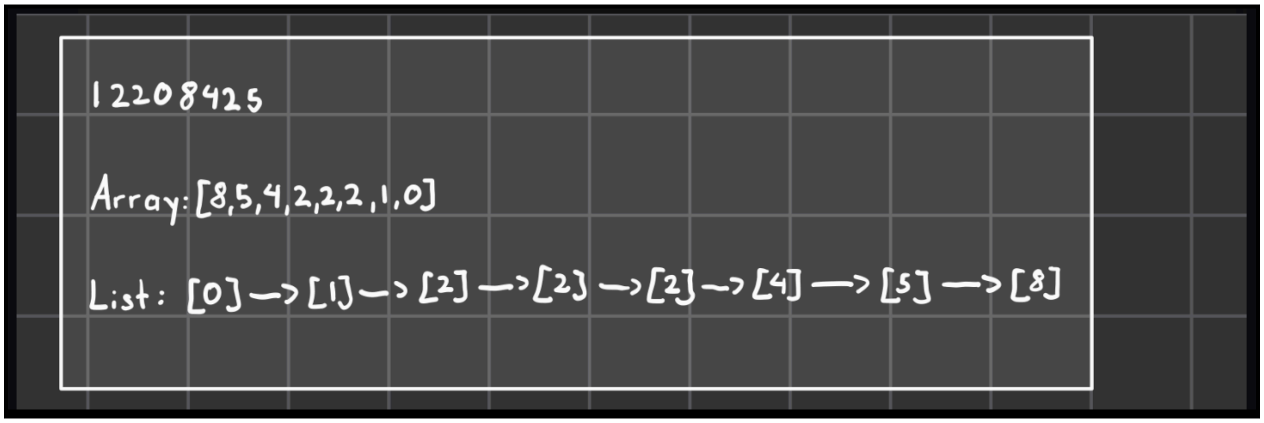
Charles Harmon

1. Array vs. Single Linked List

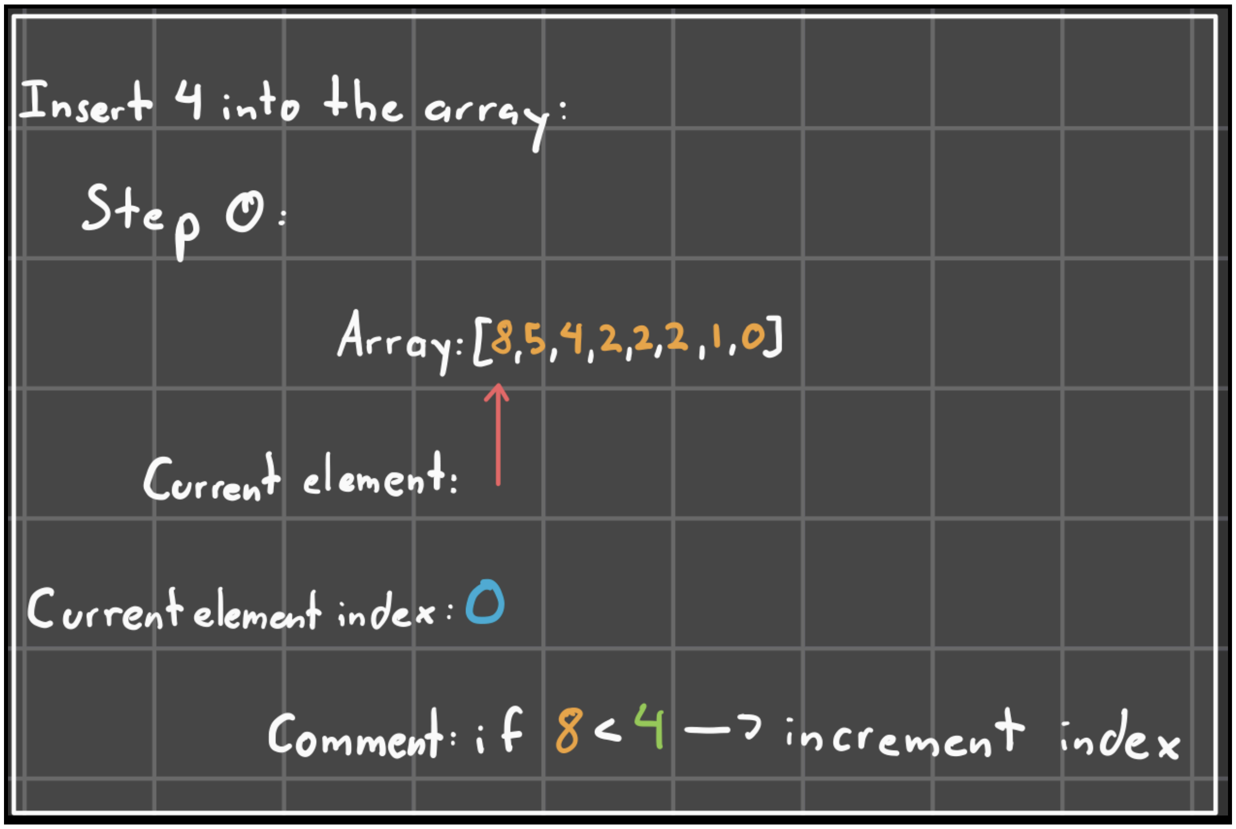


12208425

Array: [8,5,4,2,2,2,1,0]

List: [0] —> [1] —> [2] —> [2] —> [2] —> [4] —> [5] —> [8]

Array



Insert 4 into the array:

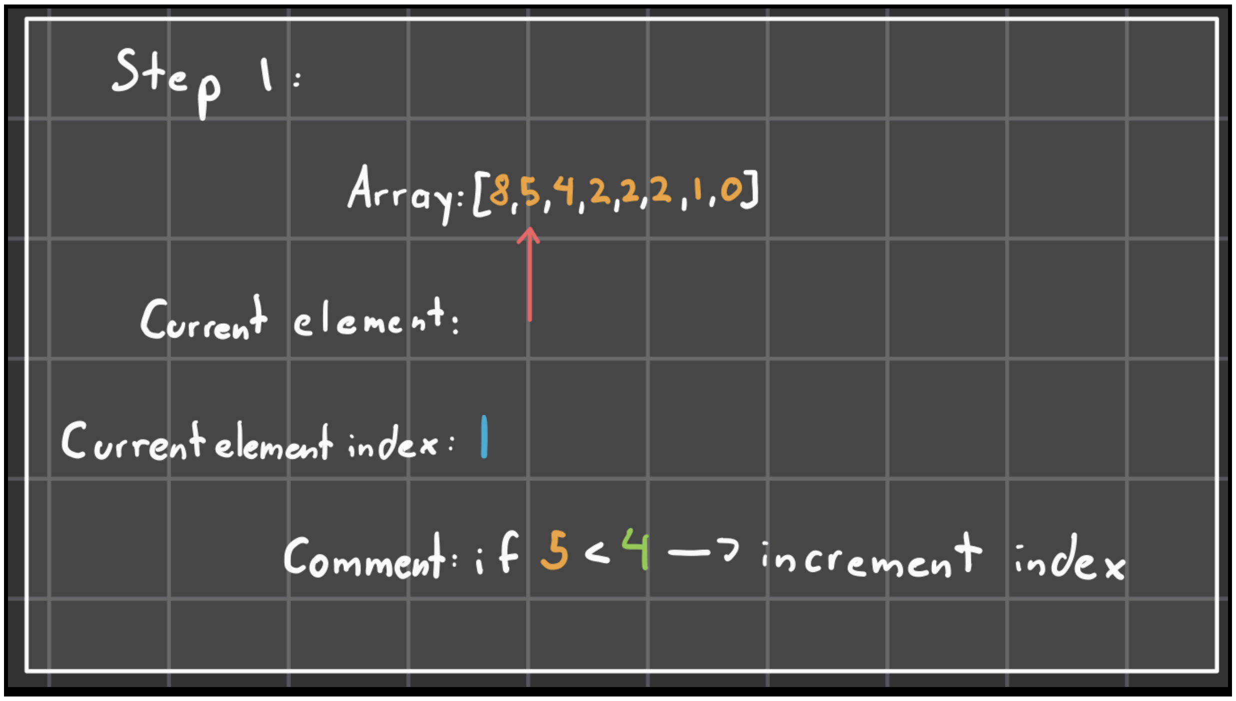
Step 0:

Array: [8,5,4,2,2,2,1,0]

Current element:

Current element index: 0

Comment: if 8 < 4 —> increment index



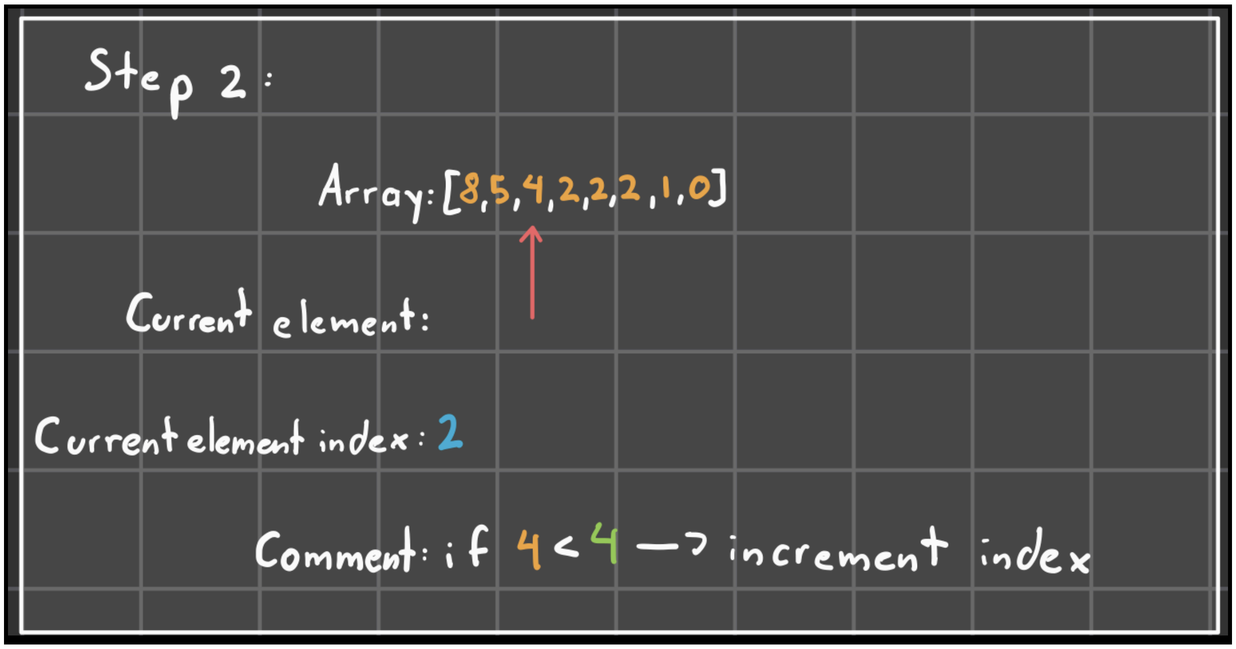
Step 1:

Array: [8,5,4,2,2,2,1,0]

Current element:

Current element index: 1

Comment: if 5 < 4 —> increment index



Step 2:

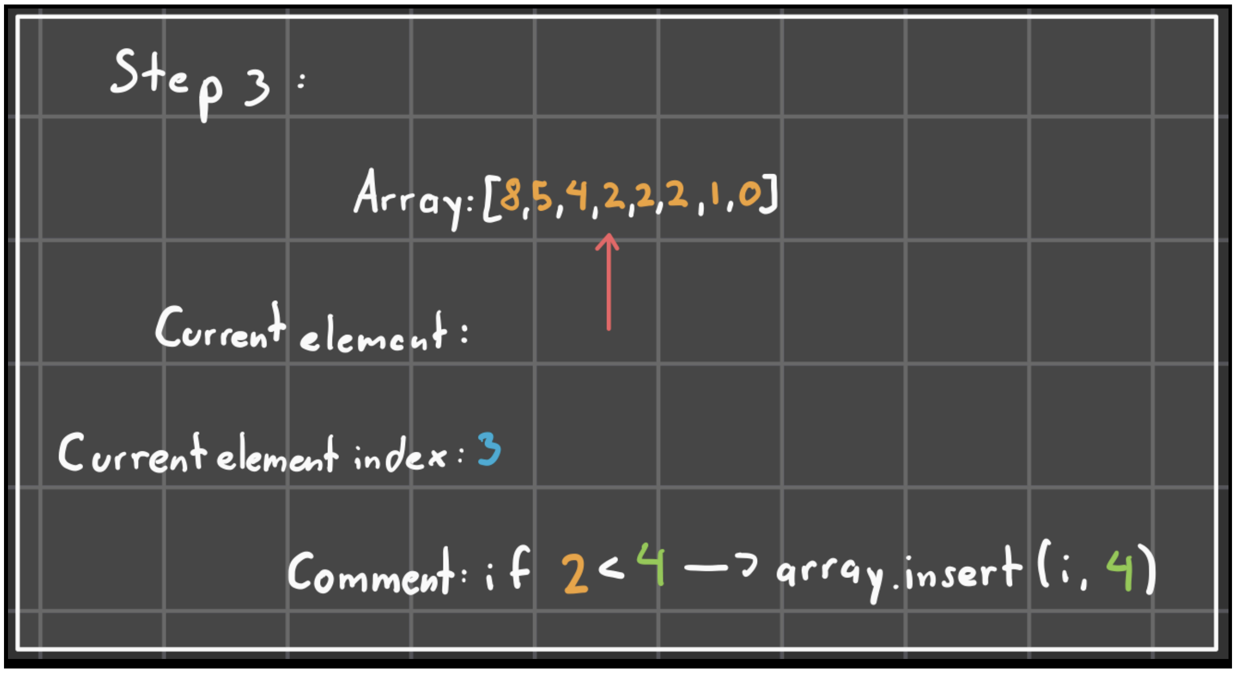
Array: [8,5,4,2,2,2,1,0]

Current element:

Current element index: 2

Comment: if 4 < 4 —> increment index

Charles Harmon



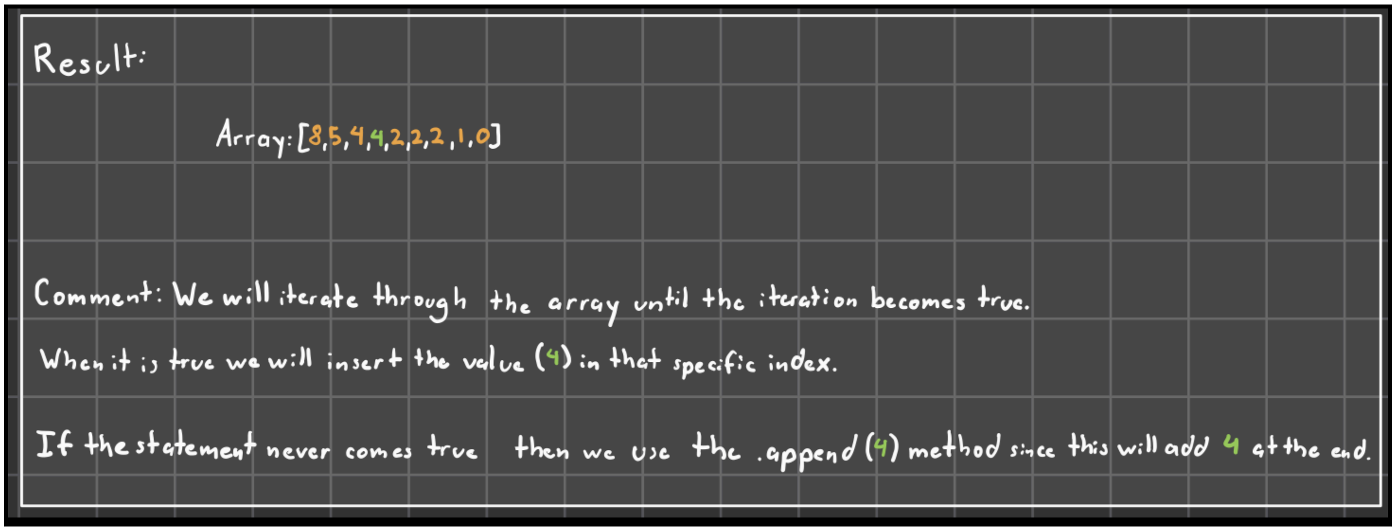
Step 3 :

Array: [8,5,4,2,2,2,1,0]

Current element :

Current element index: 3

Comment: if 2 < 4 —> array.insert(i, 4)



Result:

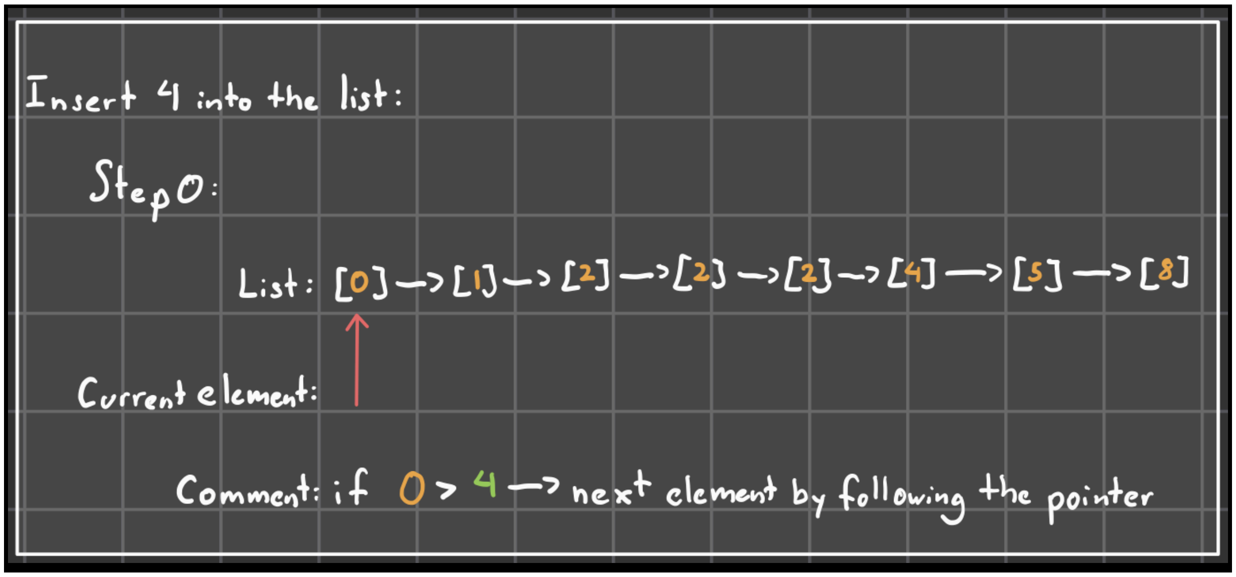
Array: [8,5,4,4,2,2,2,1,0]

Comment: We will iterate through the array until the iteration becomes true.

When it is true we will insert the value (4) in that specific index.

If the statement never comes true then we use the .append(4) method since this will add 4 at the end.

Singly Linked List

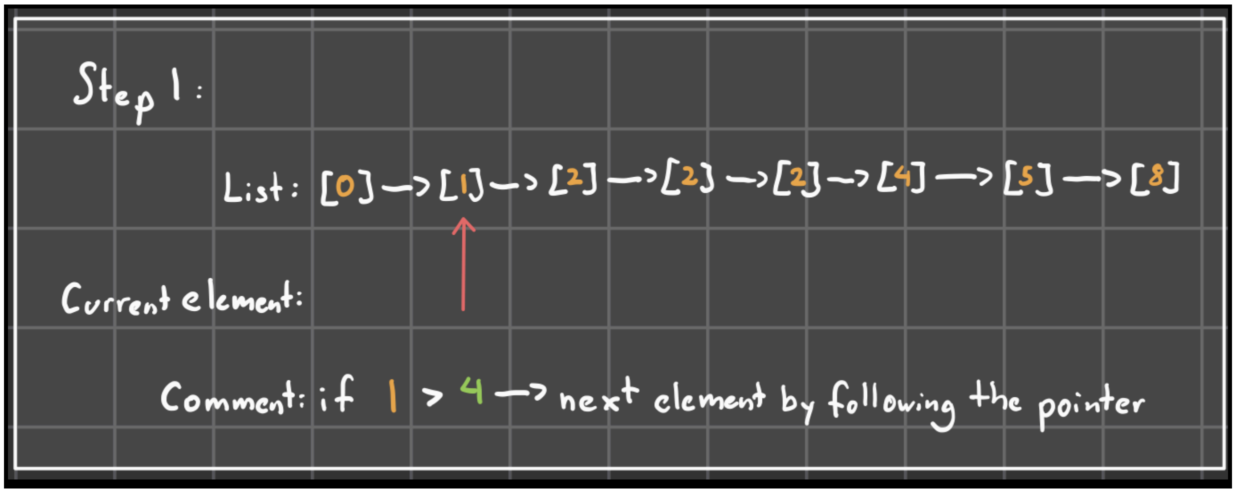

Insert 4 into the list:

Step 0:

List: [0] —> [1] —> [2] —> [2] —> [2] —> [4] —> [5] —> [8]

Current element:

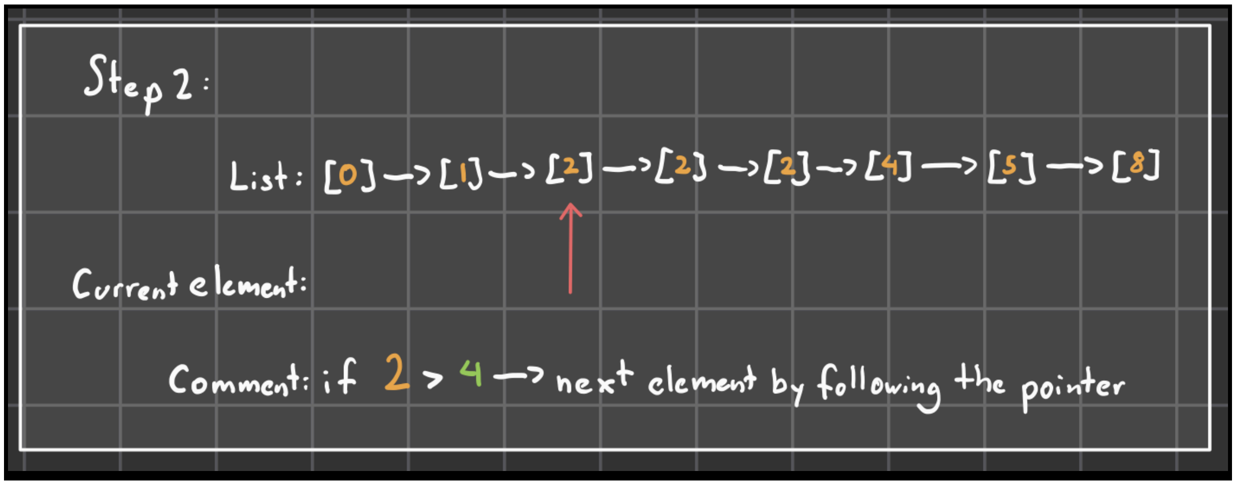
Comment: if 0 > 4 —> next element by following the pointer

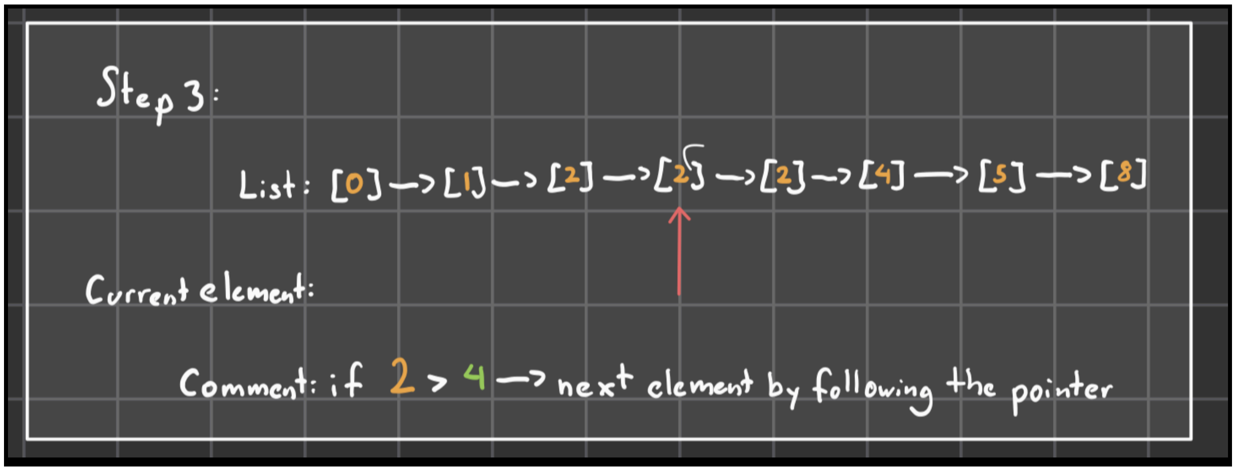

Step 1:

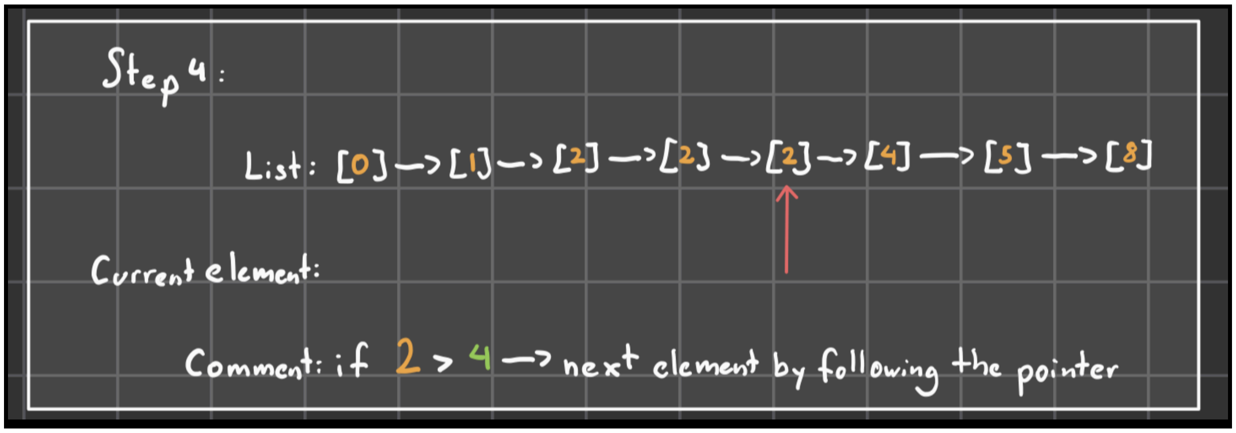
List: [0] —> [1] —> [2] —> [2] —> [2] —> [4] —> [5] —> [8]

Current element:

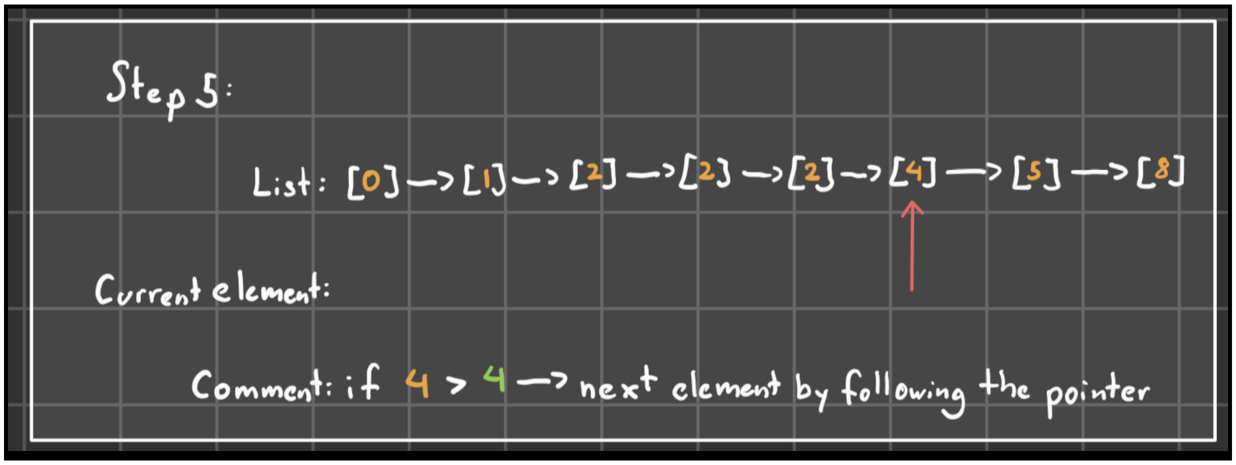
Comment: if 1 > 4 —> next element by following the pointer


Step 2:
List: [0] → [1] → [2] → [2] → [2] → [4] → [5] → [8]
Current element:
Comment: if 2 > 4 → next element by following the pointer


Step 3:
List: [0] → [1] → [2] → [2] → [2] → [4] → [5] → [8]
Current element:
Comment: if 2 > 4 → next element by following the pointer


Step 4:
List: [0] → [1] → [2] → [2] → [2] → [4] → [5] → [8]
Current element:
Comment: if 2 > 4 → next element by following the pointer

Charles Harmon

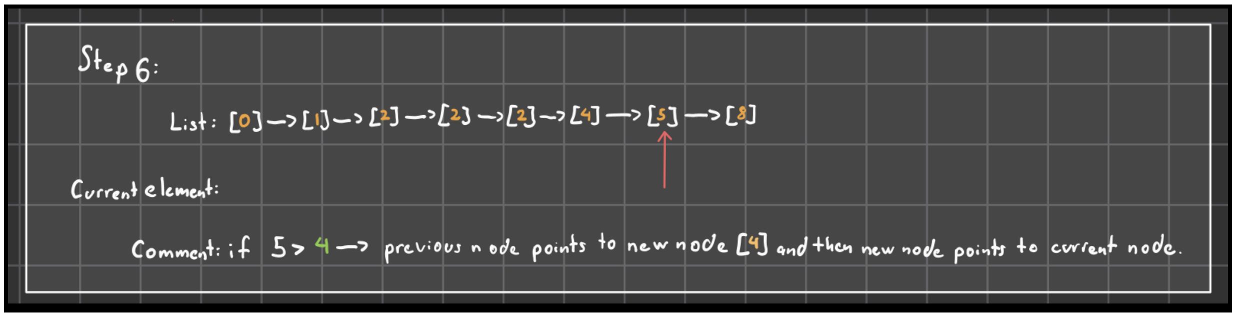


Step 5:

List: [0] —> [1] —> [2] —> [2] —> [2] —> [4] —> [5] —> [8]

Current element:

Comment: if 4 > 4 —> next element by following the pointer

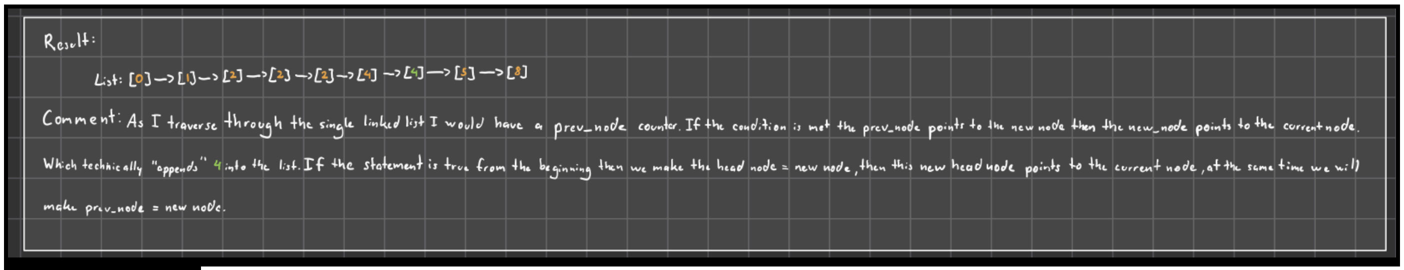


Step 6:

List: [0] —> [1] —> [2] —> [2] —> [2] —> [4] —> [5] —> [8]

Current element:

Comment: if 5 > 4 —> previous node points to new node [4] and then new node points to current node.



Result:

List: [0] —> [1] —> [2] —> [2] —> [2] —> [4] —> [4] —> [5] —> [8]

Comment: As I traverse through the single linked list I would have a prev_node counter. If the condition is met the prev_node points to the new node then the new_node points to the current node. Which technically "appends" 4 into the list. If the statement is true from the beginning then we make the head node = new node, then this new head node points to the current node, at the same time we will make prev_node = new node.

Comment:
As I traverse through the single linked list I would have a prev_node counter. If the condition is met the prev_node points to the new_node . The new_node points to the current_node. Which technically "appends" 4 into the list. If the statement is true from the beginning then we make the head_node = new_node, then this new_head_node points to the current_node, but at the same time we will make prev_node = new_node.