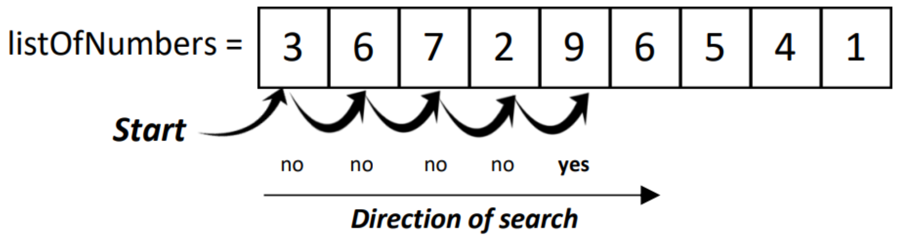
KNOWLEDGE ORGANISER :: SEARCHING ALGORITHMS

LINEAR SEARCH

- Start with the first item in the list and compare it to the criteria
- If no match is found, move on to the next item in the list and compare
- Repeat these steps until you reach the end of the list

```
function linearsearch(listOfNum, item)
    index = 1
    i - 0
    found = false
    while I < length(listOfNum) and not found
        if listOfNum[i] == item then
             index = 1
             found = true
        end if
        i = i + 1
    end while
    return index
end function
```

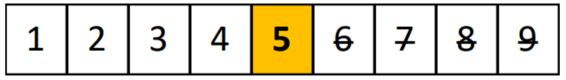


BINARY SEARCH

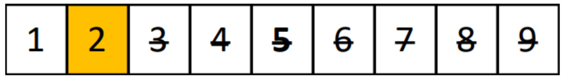
- Sort the list into order
- Split the list in half to find the middle value
- The middle value is n + 1 /2
- Compare the criteria to the middle value – is there a match?
- If no, if the criteria greater than the middle value?
- If yes, take the top half of the list, otherwise take the bottom half of the list
- Repeat the steps again

```
function binarysearch(listOfNum, item)
    index = -1 first = 0 found = False
    last = len(listOfNum) - 1
    while first <= last AND found = False
        midpoint = ((first + last) DIV 2)
        if listOfNum[midpoint] = item then
            found = True, index = midpoint
        else
            if listOfNum[midpoint] < item then
                first = midpoint + 1
            else
                last = midpoint - 1
            end if
        end if
    end while
    return index #index = -1 if key not found
end function
```

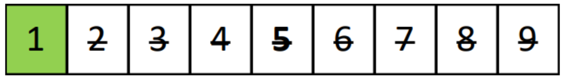
When looking for the number **1** in *listOfNumbers[]*



Find midpoint which is 5. 1 is less than 5 so we discard the second half of the list.



Find the midpoint of the new list which is 2 (using DIV which rounds down). 1 is less than 2 so discard second half.



Find midpoint of new list which is 1. Match against item we are searching for which is 1. ITEM FOUND!