Learning Aims

Understand the need for standard searching algorithms

- Binary search and linear search
- Bubble sort

Implement binary and linear search.

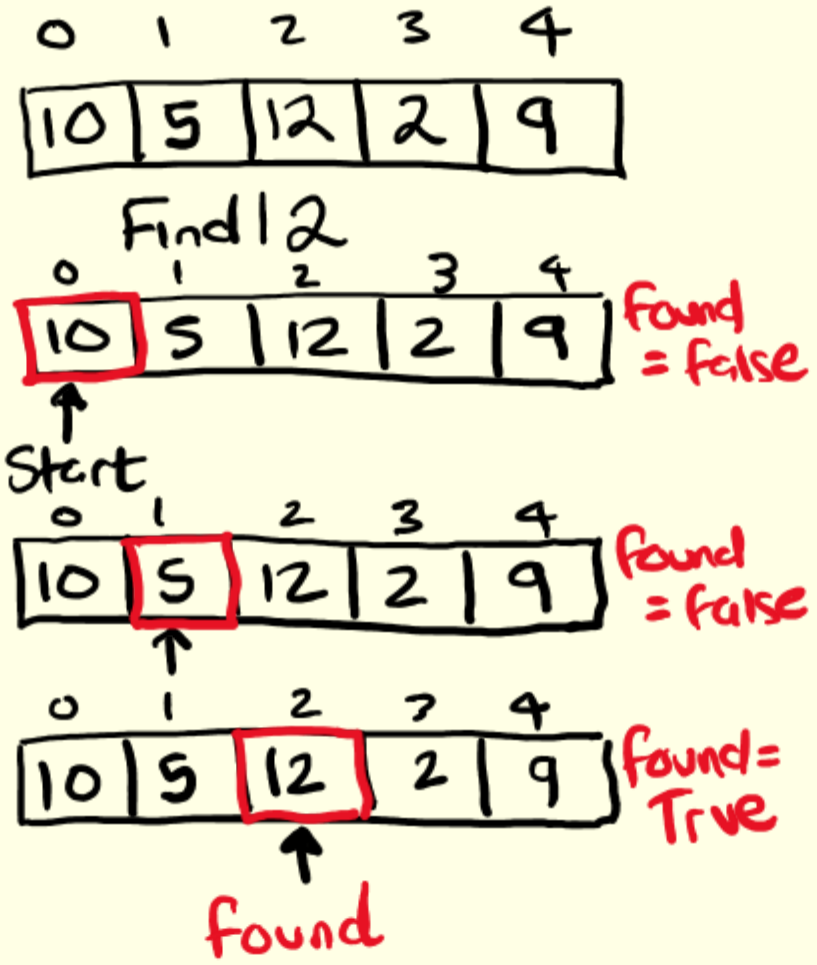
Searching Algorithms

The **linear search algorithm** looks through each item in a data list until a match is found. List does not need to be sorted and better with smaller lists

The **binary search algorithm** needs a sorted data list, then keeps splitting it in half until a match is found. Must be sorted. Better with large datasets.

Linear Search

- Starting at the first element
- Each item is checked
- until value is found
- or end of list reached and not found

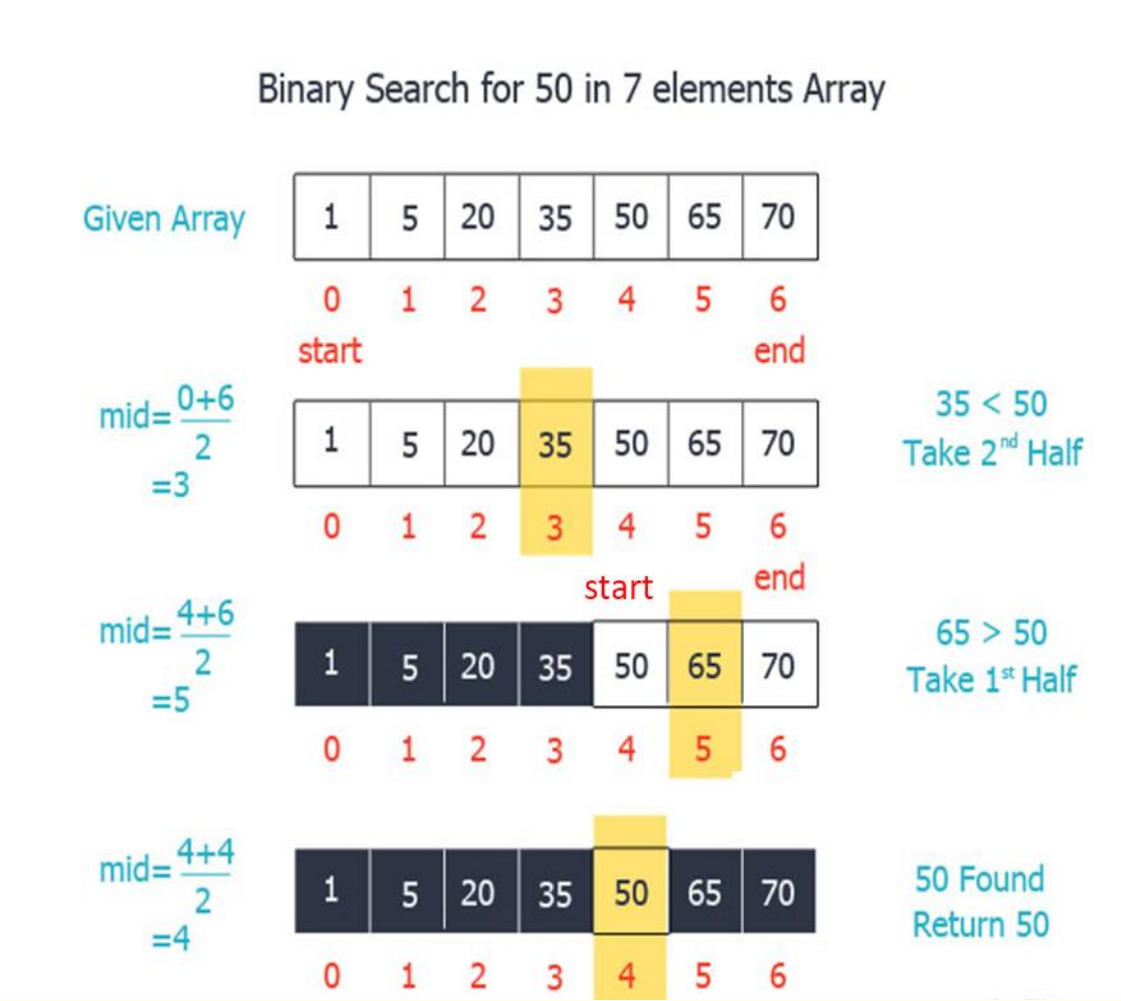


Linear Search Algorithm

```
alist = [‘A’, ‘F’, ‘B’, ‘E’, ‘D’,’G’,’C’]
position = 0
found = False
item =input()
WHILE position < len(alist) AND found == False
    IF  item == alist[position] then
            print (“Item found”)
            found = True
     ELSE
            position = position + 1
     ENDIF
END WHILE
If found == False then
     print(“Item not found”)
ENDIF
```

Binary Search Algorithm

- Binary can only search a list that has been **sorted.**
- It operates by finding the **midpoint**.
- It can then discard the side without the target.
- This finds the midpoint in the side with the target and discards half.
- This process continues until the target has been found.
- Binary Searching is quicker than linear.



Binary search algorithm

```
alist = [1,2,5,7,11,14]
item= input()
found = False
first = 0
last = len(alist) – 1
WHILE found = False AND first <= last
    midPoint = (first + last) DIV 2
    IF item ==alist[midpoint] then
          print (“item found at location”, midpoint)
          found = True
    ELSE
          IF item < alist[midpoint] then
                last = midpoint - 1
          ELSE
                first = midpoint + 1
          END IF
      END IF
END WHILE
if found == False then
       print(“Item not found”)
```

DIV is used so that it returns a whole number.
It will round down.
In python you use //

Model Answer

Show how a binary search would be performed on the array shown in Fig. 4.2 to find the value ‘duck’. [3]

midpoint = 0+8
= 4

wolf	monkey	lion	iguana	goat	giraffe	frog	elephant	duck

midpoint = 5+8
= 6

wolf	monkey	lion	iguana	goat	giraffe	frog	elephant	duck

Midpoint = 7+8
= 7

wolf	monkey	lion	iguana	goat	giraffe	frog	elephant	duck

midpoint = 8+8
= 8

wolf	monkey	lion	iguana	goat	giraffe	frog	elephant	duck

Time Efficiency

Worst Case - Time complexity of linear search is O(n) - number of elements in a list or array increase, the number of comparisons also increases linearly for a linear search algorithm

Binary search has Worst case time complexity of O(log n) – meaning it runs in logarithmic time in the worst case. Binary search is faster than linear search except for small arrays.