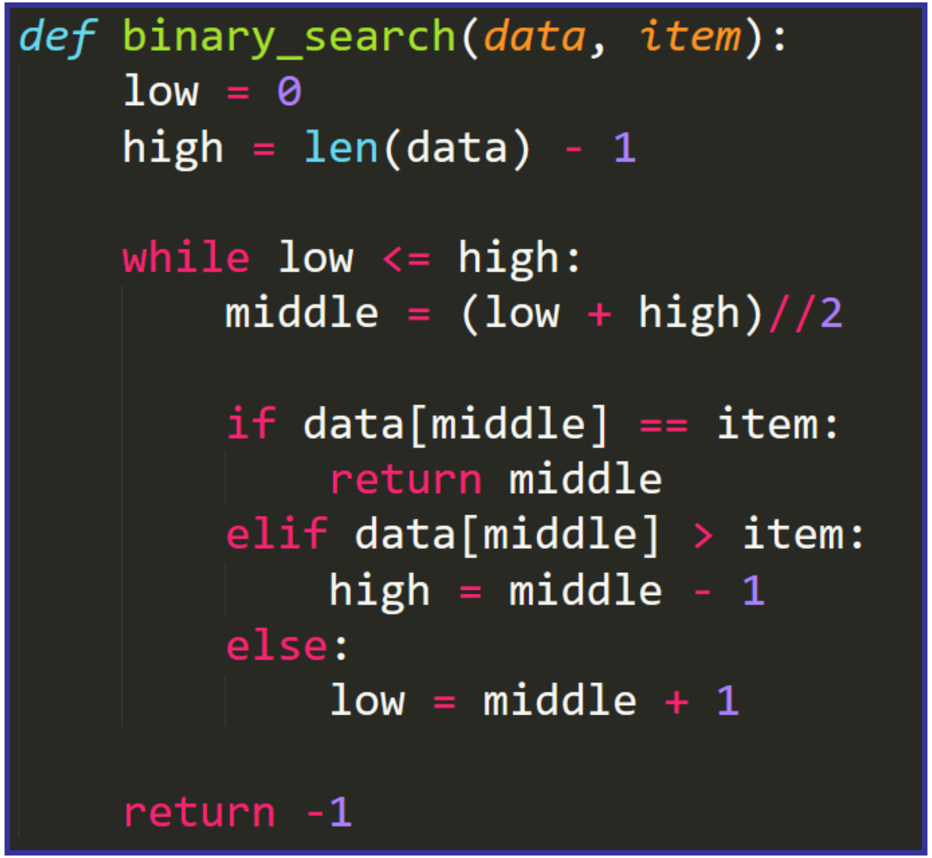
Algorithm

Binary Search

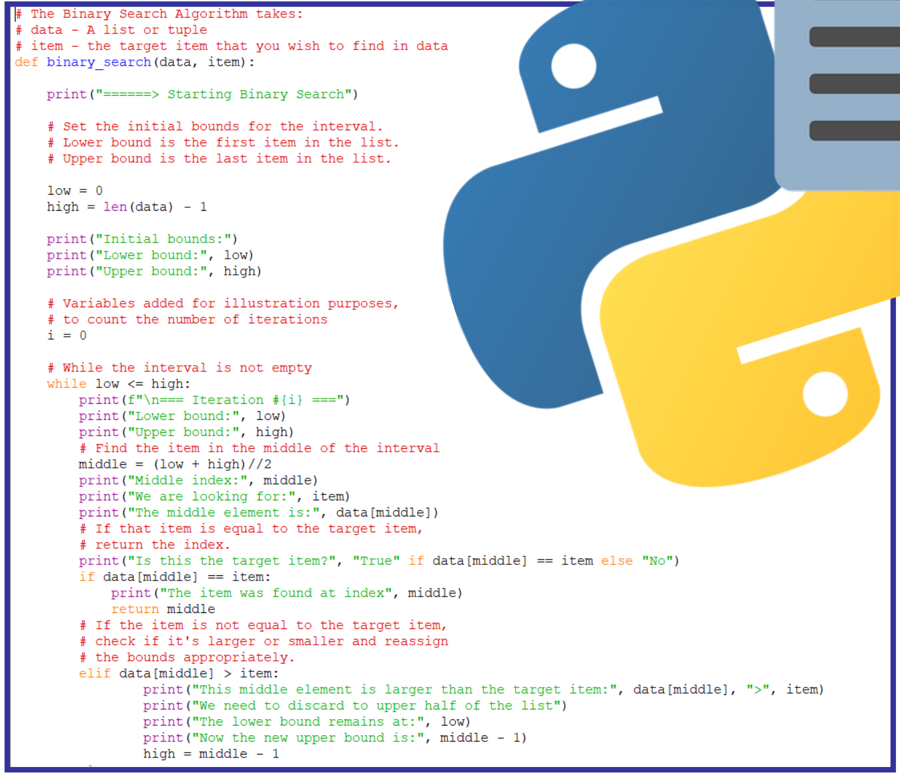
Code Walkthrough

Binary Search

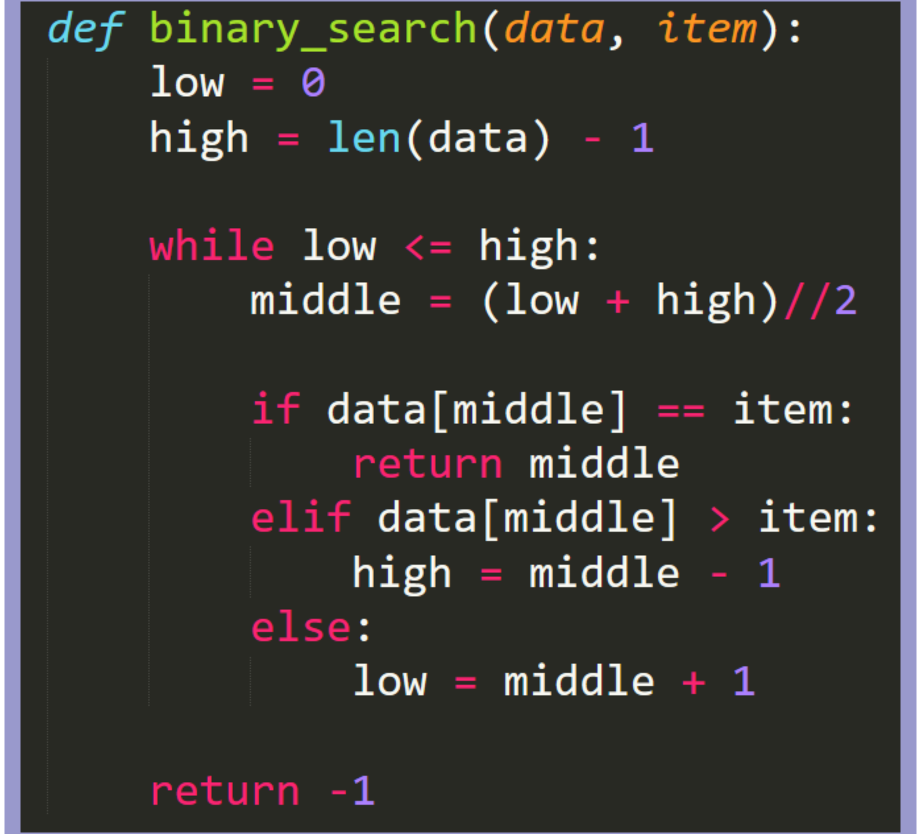


```python
def binary_search(data, item):
    low = 0
    high = len(data) - 1

    while low <= high:
        middle = (low + high)//2

        if data[middle] == item:
            return middle
        elif data[middle] > item:
            high = middle - 1
        else:
            low = middle + 1

    return -1
```

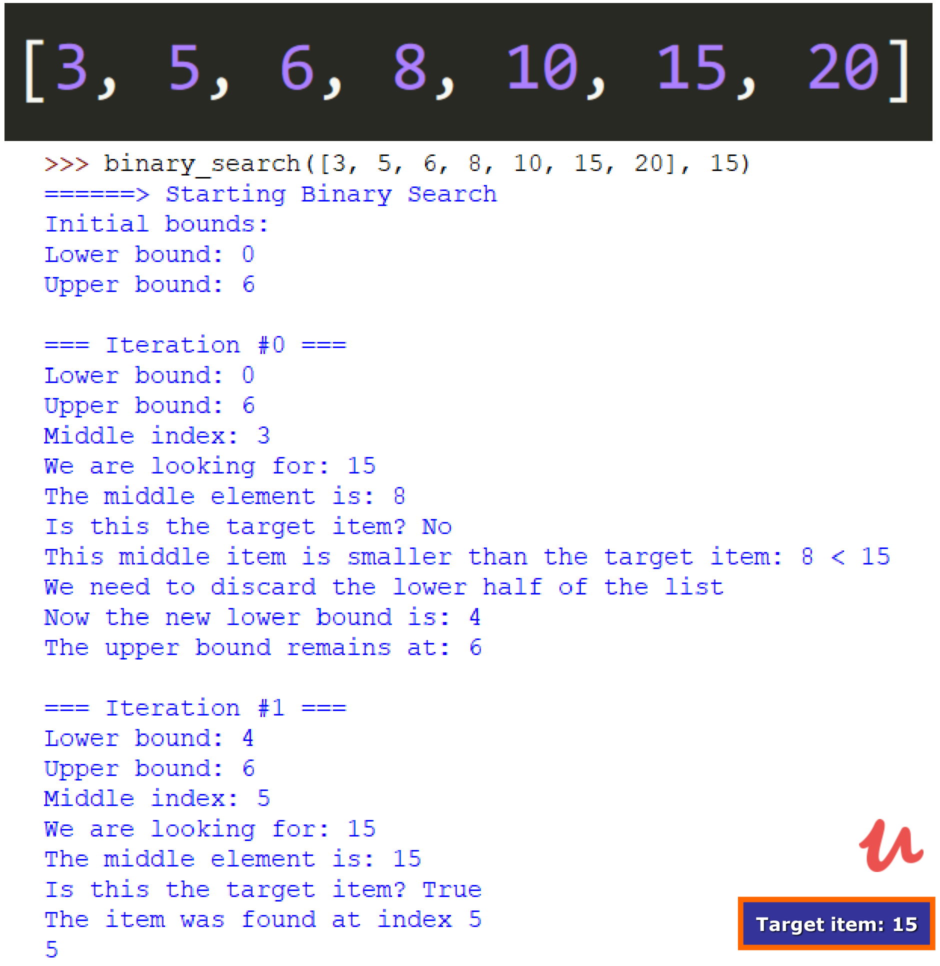
Binary Search



```python
# The Binary Search Algorithm takes:
# data - A list or tuple
# item - the target item that you wish to find in data
def binary_search(data, item):

    print("======> Starting Binary Search")

    # Set the initial bounds for the interval.
    # Lower bound is the first item in the list.
    # Upper bound is the last item in the list.

    low = 0
    high = len(data) - 1

    print("Initial bounds:")
    print("Lower bound:", low)
    print("Upper bound:", high)

    # Variables added for illustration purposes,
    # to count the number of iterations
    i = 0

    # While the interval is not empty
    while low <= high:
        print(f"\n=== Iteration #{i} ===")
        print("Lower bound:", low)
        print("Upper bound:", high)
        # Find the item in the middle of the interval
        middle = (low + high)//2
        print("Middle index:", middle)
        print("We are looking for:", item)
        print("The middle element is:", data[middle])
        # If that item is equal to the target item,
        # return the index.
        print("Is this the target item?", "True" if data[middle] == item else "No")
        if data[middle] == item:
            print("The item was found at index", middle)
            return middle
        # If the item is not equal to the target item,
        # check if it's larger or smaller and reassign
        # the bounds appropriately.
        elif data[middle] > item:
                print("This middle element is larger than the target item:", data[middle], ">", item)
                print("We need to discard to upper half of the list")
                print("The lower bound remains at:", low)
                print("Now the new upper bound is:", middle - 1)
                high = middle - 1
```



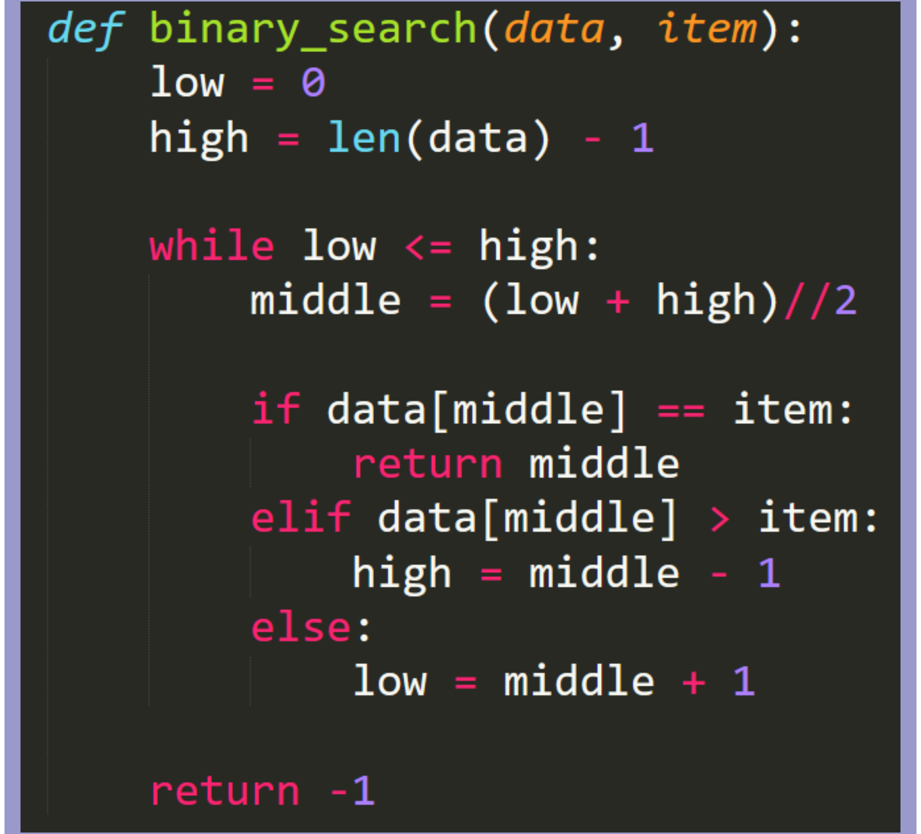
```python
def binary_search(data, item):
    low = 0
    high = len(data) - 1

    while low <= high:
        middle = (low + high)//2

        if data[middle] == item:
            return middle
        elif data[middle] > item:
            high = middle - 1
        else:
            low = middle + 1

    return -1
```



[3, 5, 6, 8, 10, 15, 20]

```
>>> binary_search([3, 5, 6, 8, 10, 15, 20], 15)
======> Starting Binary Search
Initial bounds:
Lower bound: 0
Upper bound: 6

=== Iteration #0 ===
Lower bound: 0
Upper bound: 6
Middle index: 3
We are looking for: 15
The middle element is: 8
Is this the target item? No
This middle item is smaller than the target item: 8 < 15
We need to discard the lower half of the list
Now the new lower bound is: 4
The upper bound remains at: 6

=== Iteration #1 ===
Lower bound: 4
Upper bound: 6
Middle index: 5
We are looking for: 15
The middle element is: 15
Is this the target item? True
The item was found at index 5
5
```

Target item: 15

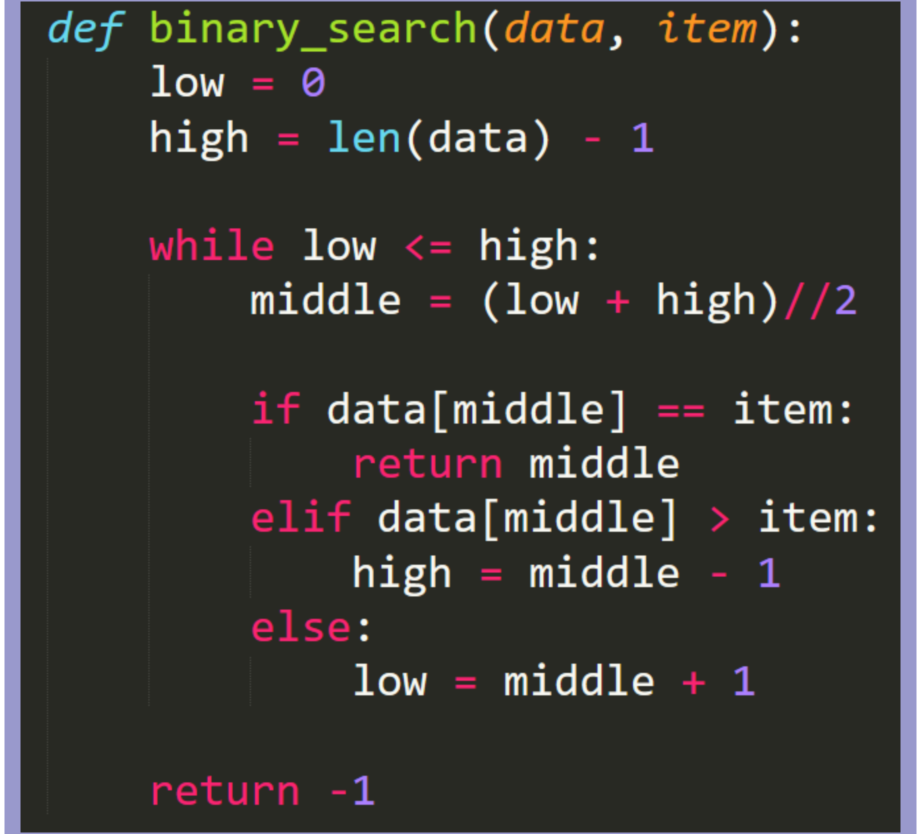


```python
def binary_search(data, item):
    low = 0
    high = len(data) - 1

    while low <= high:
        middle = (low + high)//2

        if data[middle] == item:
            return middle
        elif data[middle] > item:
            high = middle - 1
        else:
            low = middle + 1

    return -1
```

[3, 5, 6, 8, 10, 15, 20]

```
>>> binary_search([3, 5, 6, 8, 10, 15, 20], 5)
======> Starting Binary Search
Initial bounds:
Lower bound: 0
Upper bound: 6

=== Iteration #0 ===
Lower bound: 0
Upper bound: 6
Middle index: 3
We are looking for: 5
The middle element is: 8
Is this the target item? No
This middle element is greater than the target item: 8 > 5
We need to discard the upper half of the list
The lower bound remains at: 0
Now the new upper bound is: 2

=== Iteration #1 ===
Lower bound: 0
Upper bound: 2
Middle index: 1
We are looking for: 5
The middle element is: 5
Is this the target item? True
The item was found at index 1
1
```

Target item: 5

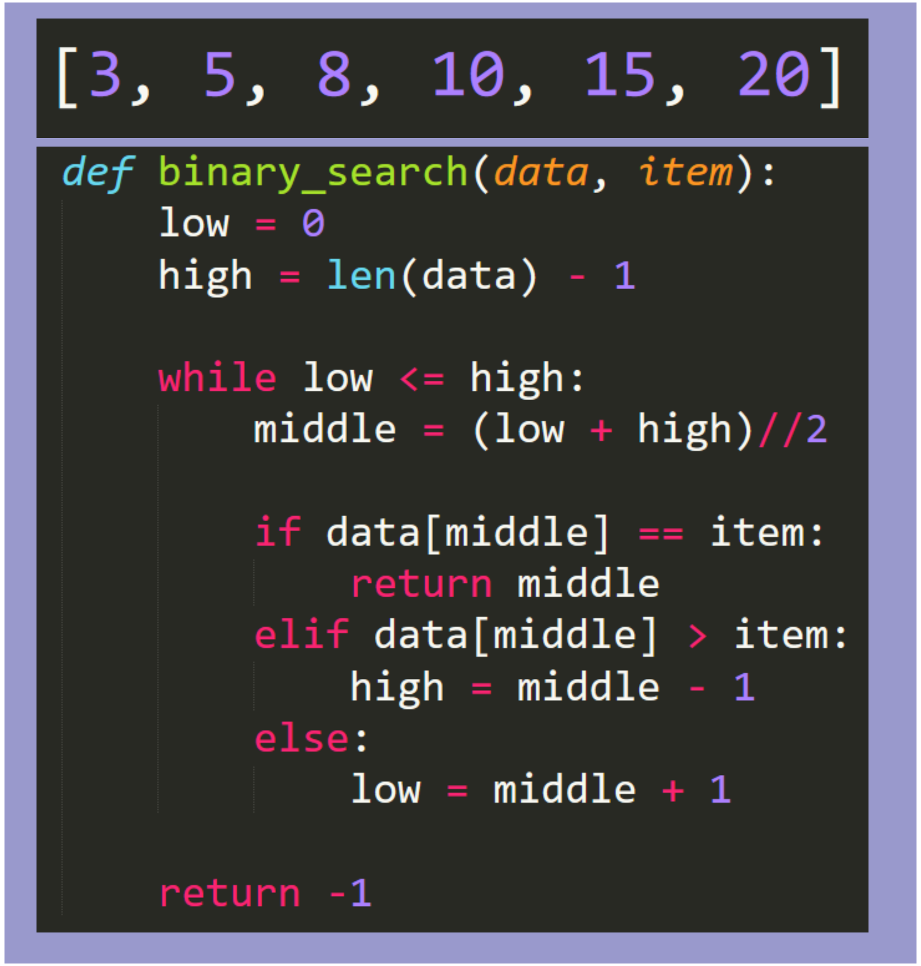


```python
def binary_search(data, item):
    low = 0
    high = len(data) - 1

    while low <= high:
        middle = (low + high)//2

        if data[middle] == item:
            return middle
        elif data[middle] > item:
            high = middle - 1
        else:
            low = middle + 1

    return -1
```

```
[3, 5, 8, 10, 15, 20]
```

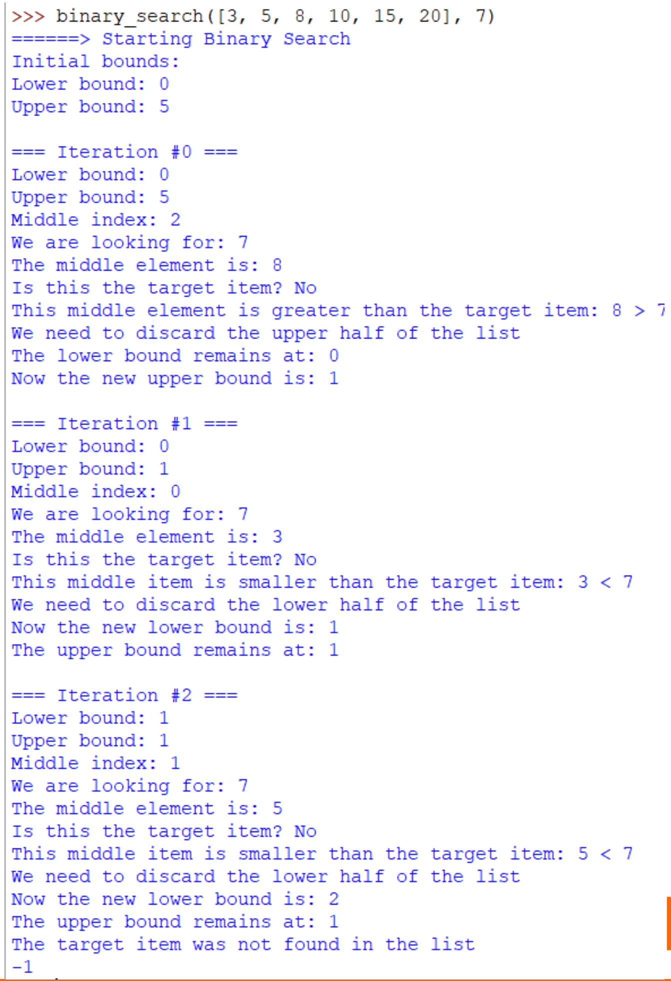
```
>>> binary_search([3, 5, 8, 10, 15, 20], 15)
======> Starting Binary Search
Initial bounds:
Lower bound: 0
Upper bound: 5

=== Iteration #0 ===
Lower bound: 0
Upper bound: 5
Middle index: 2
We are looking for: 15
The middle element is: 8
Is this the target item? No
This middle item is smaller than the target item: 8 < 15
We need to discard the lower half of the list
Now the new lower bound is: 3
The upper bound remains at: 5

=== Iteration #1 ===
Lower bound: 3
Upper bound: 5
Middle index: 4
We are looking for: 15
The middle element is: 15
Is this the target item? True
The item was found at index 4
4
```

Target item: 15



```
[3, 5, 8, 10, 15, 20]
```

```python
def binary_search(data, item):
    low = 0
    high = len(data) - 1

    while low <= high:
        middle = (low + high)//2

        if data[middle] == item:
            return middle
        elif data[middle] > item:
            high = middle - 1
        else:
            low = middle + 1

    return -1
```



```
>>> binary_search([3, 5, 8, 10, 15, 20], 7)
======> Starting Binary Search
Initial bounds:
Lower bound: 0
Upper bound: 5

=== Iteration #0 ===
Lower bound: 0
Upper bound: 5
Middle index: 2
We are looking for: 7
The middle element is: 8
Is this the target item? No
This middle element is greater than the target item: 8 > 7
We need to discard the upper half of the list
The lower bound remains at: 0
Now the new upper bound is: 1

=== Iteration #1 ===
Lower bound: 0
Upper bound: 1
Middle index: 0
We are looking for: 7
The middle element is: 3
Is this the target item? No
This middle item is smaller than the target item: 3 < 7
We need to discard the lower half of the list
Now the new lower bound is: 1
The upper bound remains at: 1

=== Iteration #2 ===
Lower bound: 1
Upper bound: 1
Middle index: 1
We are looking for: 7
The middle element is: 5
Is this the target item? No
This middle item is smaller than the target item: 5 < 7
We need to discard the lower half of the list
Now the new lower bound is: 2
The upper bound remains at: 1
The target item was not found in the list
-1
```

Target item: 7

Time to Code!