Algorithm

Binary Search

Code Walkthrough

Binary Search

```python
def binary_search(data, low, high, item):
    if low <= high:
        middle = (low + high)//2
        if data[middle] == item:
            return middle
        elif data[middle] > item:
            return binary_search(data, low, middle - 1, item)
        else:
            return binary_search(data, middle + 1, high, item)
    else:
        return -1
```

Binary Search

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

```python
def binary_search(data, low, high, item):
    if low <= high:
        middle = (low + high)//2
        if data[middle] == item:
            return middle
        elif data[middle] > item:
            return binary_search(data, low, middle - 1, item)
        else:
            return binary_search(data, middle + 1, high, item)
    else:
        return -1
```

```
>>> data = [3, 5, 6, 8, 10, 15, 20]
>>> binary_search(data, 0, len(data), 15)

===> Calling Binary Search
Lower bound: 0
Upper bound: 7
Middle index: 3
Item at middle index: 8
We are looking for: 15
Is this the item? No
The current item is smaller than the target item: 8 < 15
We need to discard the lower half of the list
The lower bound is now: 4
The upper bound remains at: 7

===> Calling Binary Search
Lower bound: 4
Upper bound: 7
Middle index: 5
Item at middle index: 15
We are looking for: 15
Is this the item? Yes
The item was found at index: 5
5
```

Target item: 15

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

```python
def binary_search(data, low, high, item):
    if low <= high:
        middle = (low + high)//2
        if data[middle] == item:
            return middle
        elif data[middle] > item:
            return binary_search(data, low, middle - 1, item)
        else:
            return binary_search(data, middle + 1, high, item)
    else:
        return -1
```

```
>>> data = [3, 5, 6, 8, 10, 15, 20]
>>> binary_search(data, 0, len(data), 5)

===> Calling Binary Search
Lower bound: 0
Upper bound: 7
Middle index: 3
Item at middle index: 8
We are looking for: 5
Is this the item? No
The current item is greater than the target item: 8 > 5
We need to discard the upper half of the list
The lower bound remains at: 0
The upper bound is now: 2

===> Calling Binary Search
Lower bound: 0
Upper bound: 2
Middle index: 1
Item at middle index: 5
We are looking for: 5
Is this the item? Yes
The item was found at index: 1
1
```

Target item: 5

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

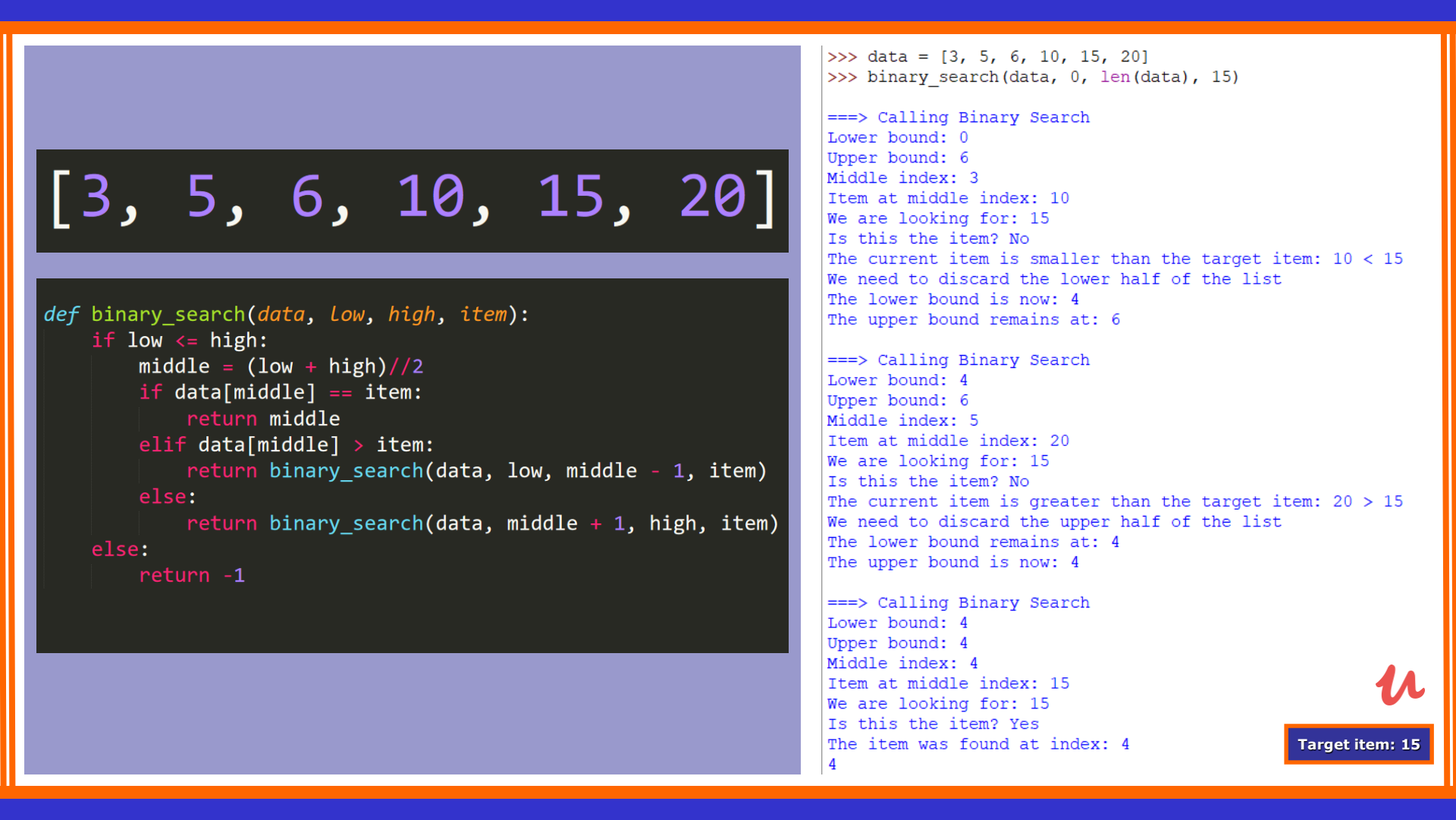
```python
def binary_search(data, low, high, item):
    if low <= high:
        middle = (low + high)//2
        if data[middle] == item:
            return middle
        elif data[middle] > item:
            return binary_search(data, low, middle - 1, item)
        else:
            return binary_search(data, middle + 1, high, item)
    else:
        return -1
```

```
>>> data = [3, 5, 6, 10, 15, 20]
>>> binary_search(data, 0, len(data), 15)

===> Calling Binary Search
Lower bound: 0
Upper bound: 6
Middle index: 3
Item at middle index: 10
We are looking for: 15
Is this the item? No
The current item is smaller than the target item: 10 < 15
We need to discard the lower half of the list
The lower bound is now: 4
The upper bound remains at: 6

===> Calling Binary Search
Lower bound: 4
Upper bound: 6
Middle index: 5
Item at middle index: 20
We are looking for: 15
Is this the item? No
The current item is greater than the target item: 20 > 15
We need to discard the upper half of the list
The lower bound remains at: 4
The upper bound is now: 4

===> Calling Binary Search
Lower bound: 4
Upper bound: 4
Middle index: 4
Item at middle index: 15
We are looking for: 15
Is this the item? Yes
The item was found at index: 4
4
```



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

```python
def binary_search(data, low, high, item):
    if low <= high:
        middle = (low + high)//2
        if data[middle] == item:
            return middle
        elif data[middle] > item:
            return binary_search(data, low, middle - 1, item)
        else:
            return binary_search(data, middle + 1, high, item)
    else:
        return -1
```

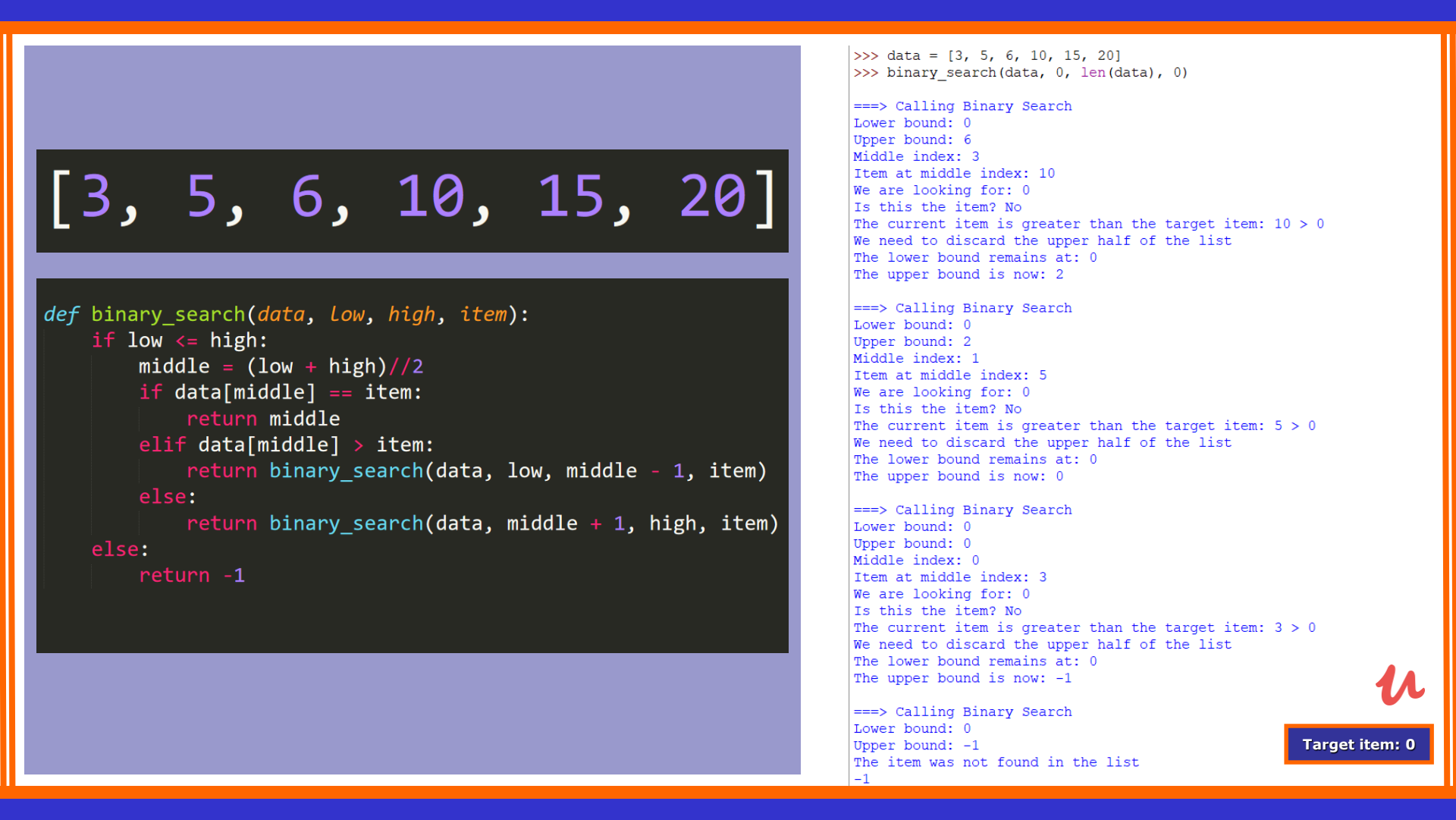
```
>>> data = [3, 5, 6, 10, 15, 20]
>>> binary_search(data, 0, len(data), 0)

===> Calling Binary Search
Lower bound: 0
Upper bound: 6
Middle index: 3
Item at middle index: 10
We are looking for: 0
Is this the item? No
The current item is greater than the target item: 10 > 0
We need to discard the upper half of the list
The lower bound remains at: 0
The upper bound is now: 2

===> Calling Binary Search
Lower bound: 0
Upper bound: 2
Middle index: 1
Item at middle index: 5
We are looking for: 0
Is this the item? No
The current item is greater than the target item: 5 > 0
We need to discard the upper half of the list
The lower bound remains at: 0
The upper bound is now: 0

===> Calling Binary Search
Lower bound: 0
Upper bound: 0
Middle index: 0
Item at middle index: 3
We are looking for: 0
Is this the item? No
The current item is greater than the target item: 3 > 0
We need to discard the upper half of the list
The lower bound remains at: 0
The upper bound is now: -1

===> Calling Binary Search
Lower bound: 0
Upper bound: -1
The item was not found in the list
-1
```



Time to Practice