Algorithm

Binary Search

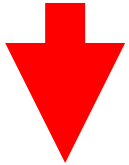
Visual Walkthrough

Binary Search

- **Very Efficient** search algorithm.
- List or Tuple **has to be sorted**!
- Eliminates **half the options** per iteration.
- Also called “Half-interval Search”.
- Can be used for large lists and tuples.

Binary Search

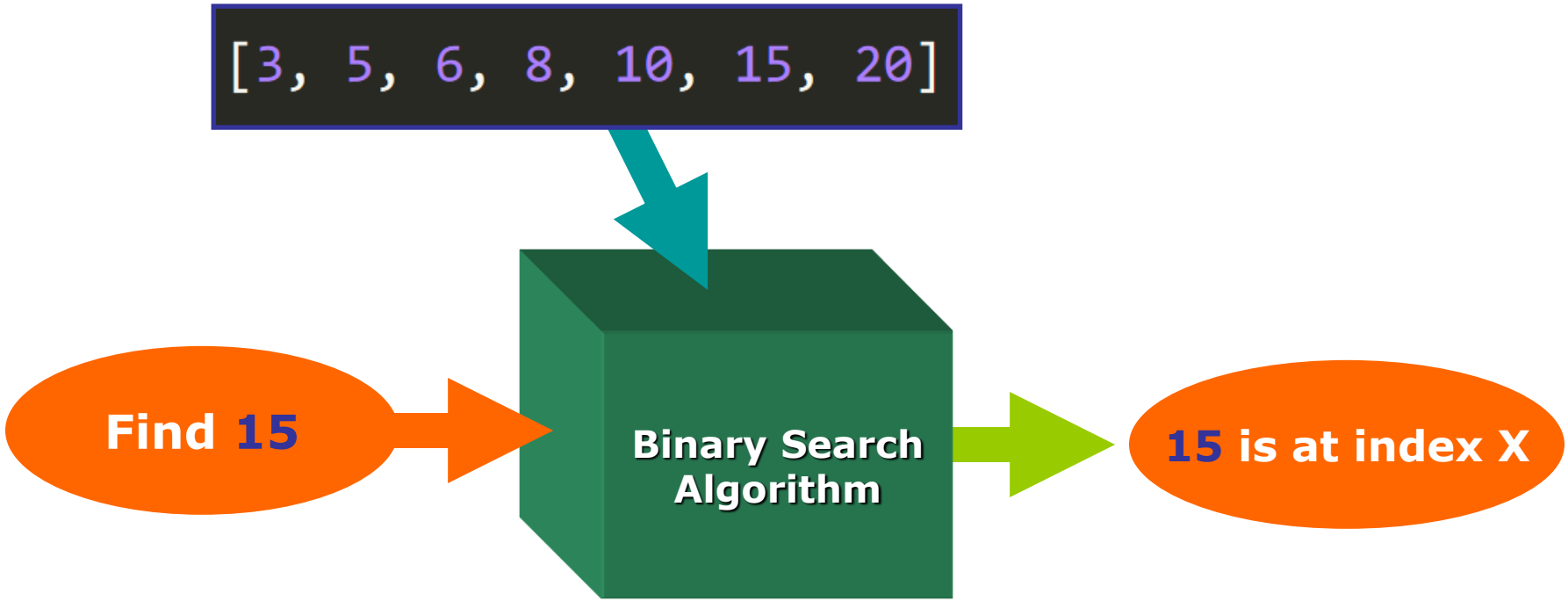
```
[5, 8, 3, 6, 15, 20, 10]
```



Sort

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


Binary Search



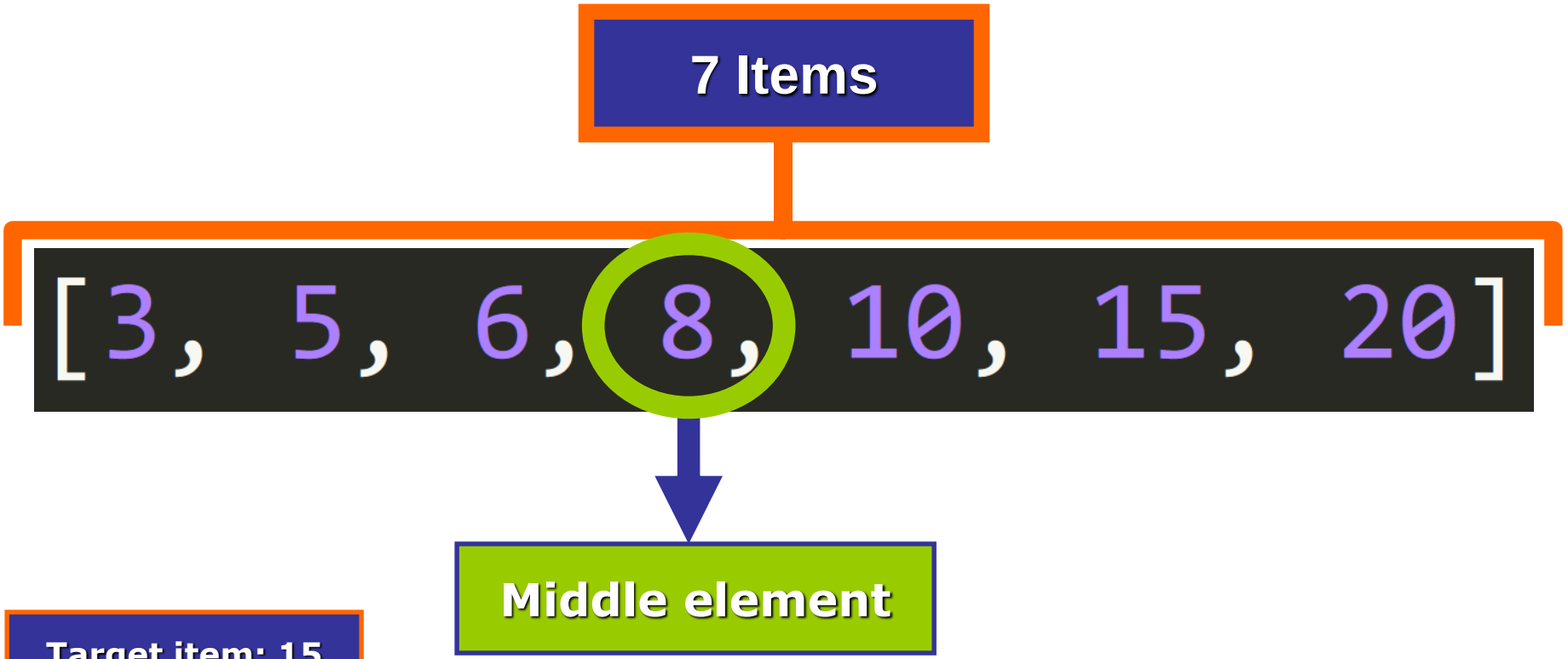
Binary Search

7 Items

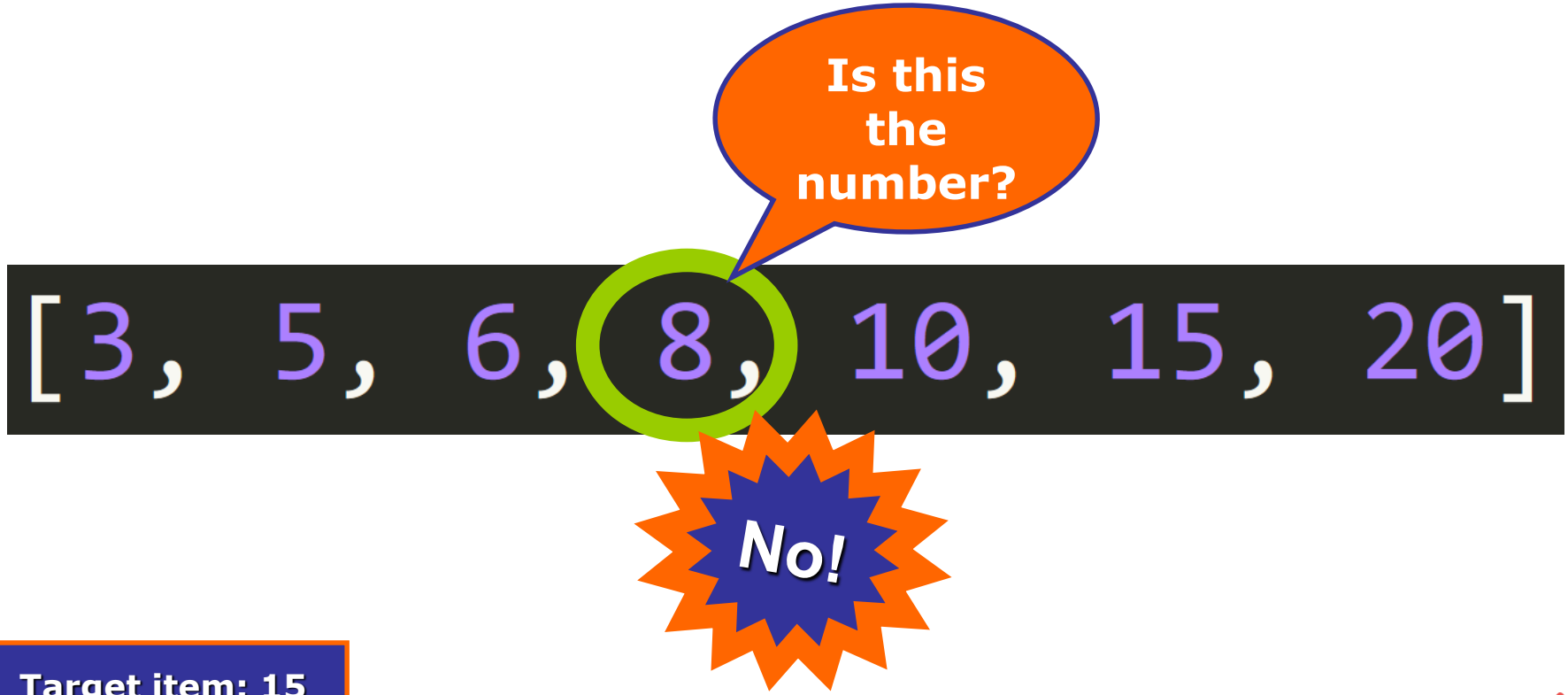
```
[3, 5, 6, 8, 10, 15, 20]
```

Target item: 15

Binary Search



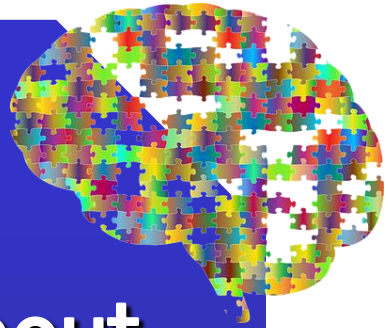
Binary Search



Binary Search

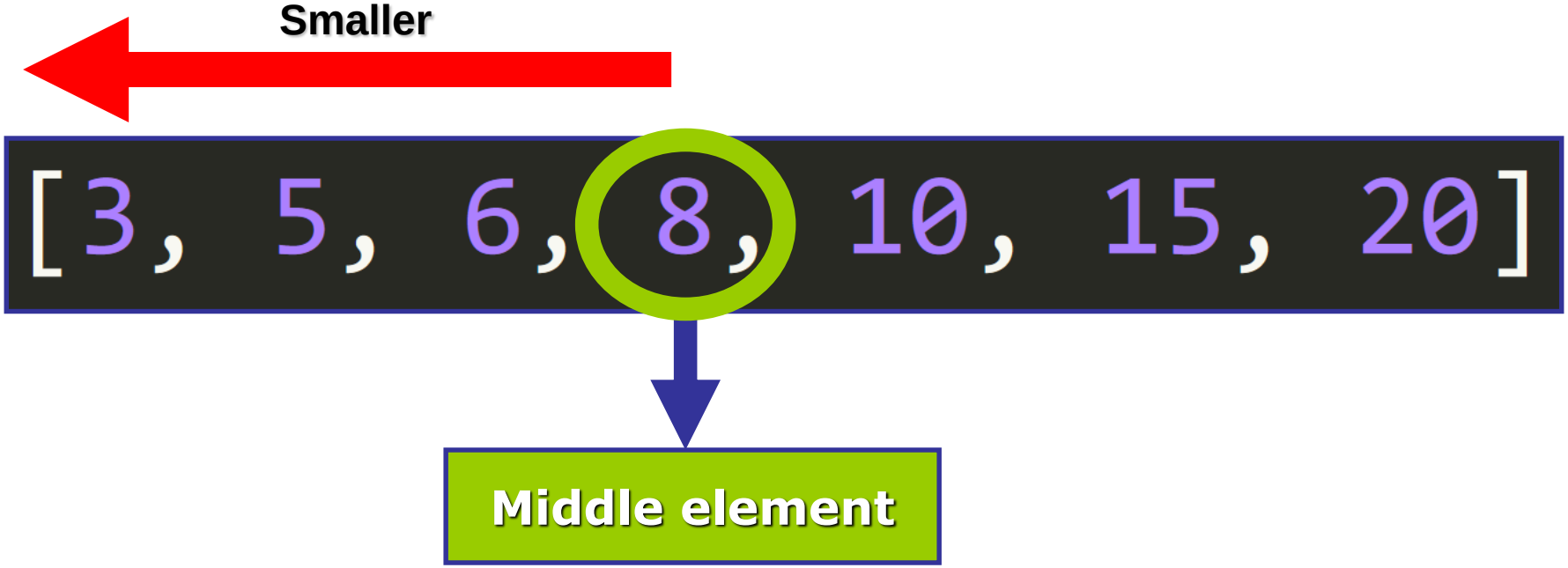
15 > 8

Target item: 15



Let's think about... where to search

Binary Search



Binary Search

Smaller than target item

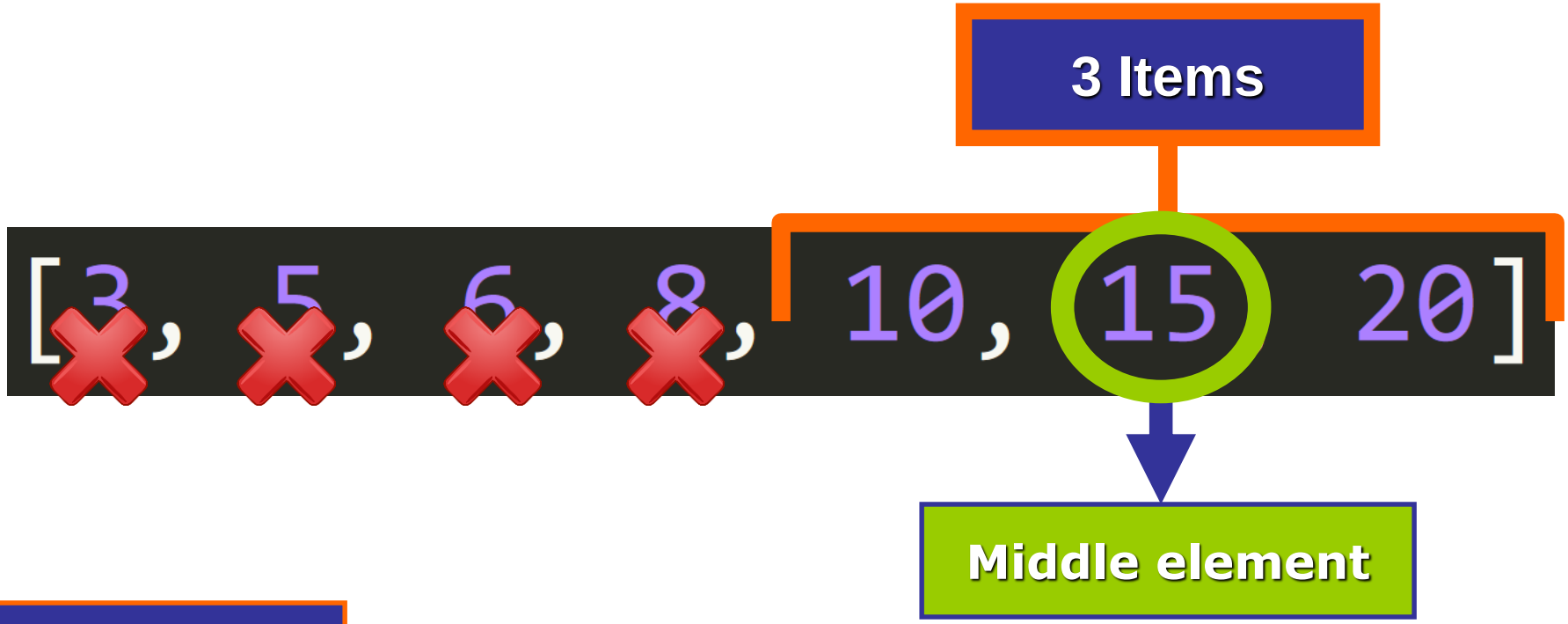
Target item: 15

Binary Search

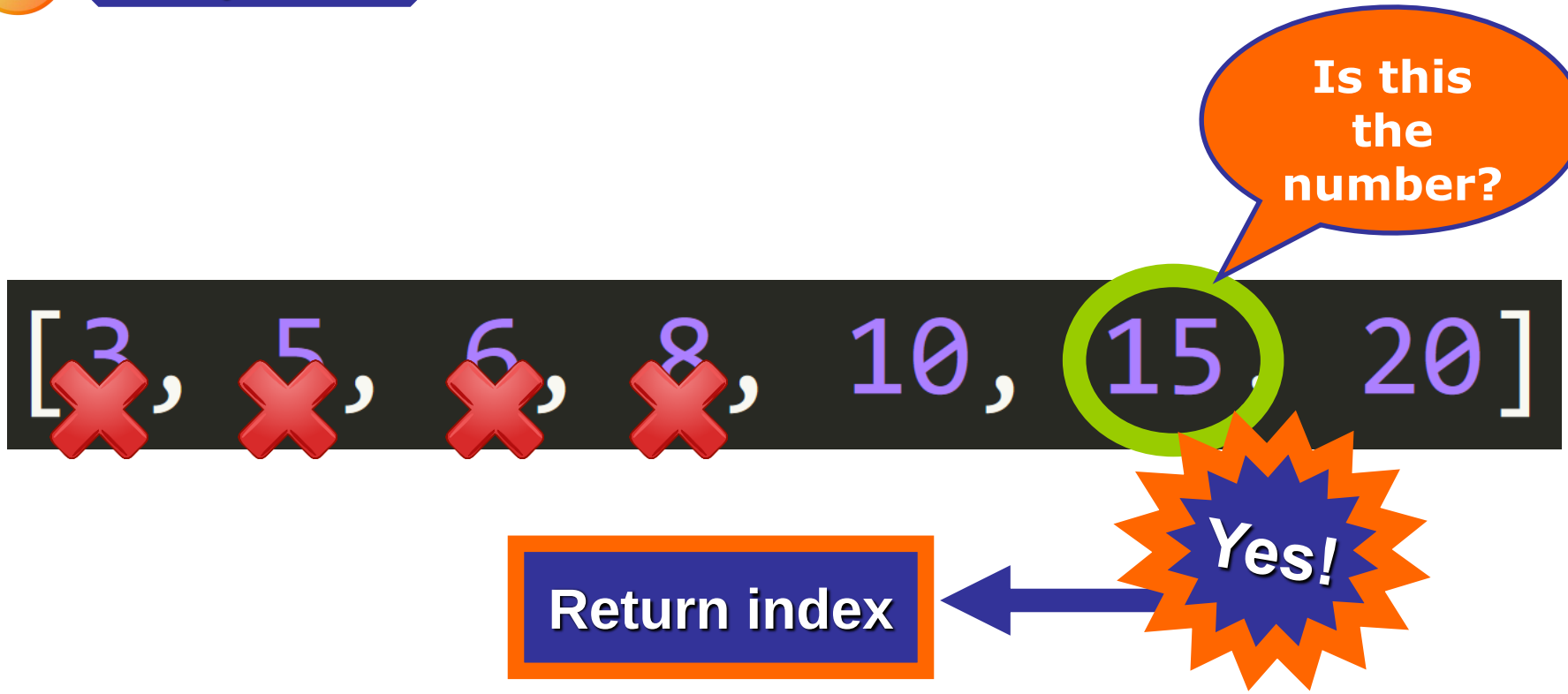
The item could be here

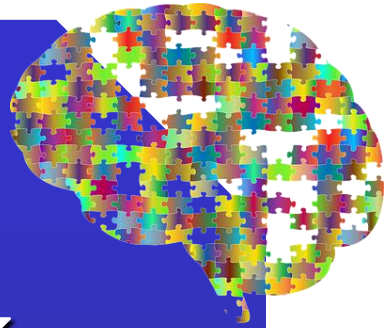
Target item: 15

Binary Search



Binary Search





Let's think...
if it's smaller

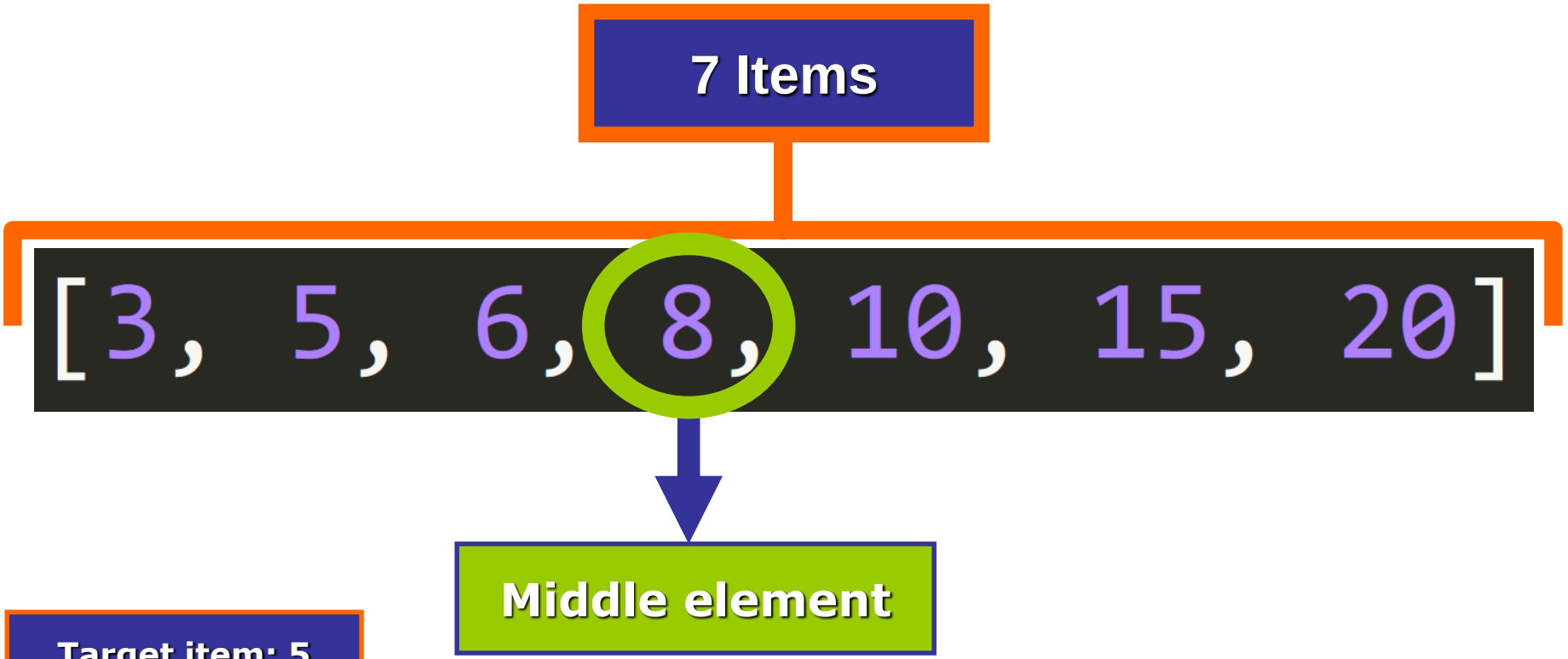
Binary Search

7 Items

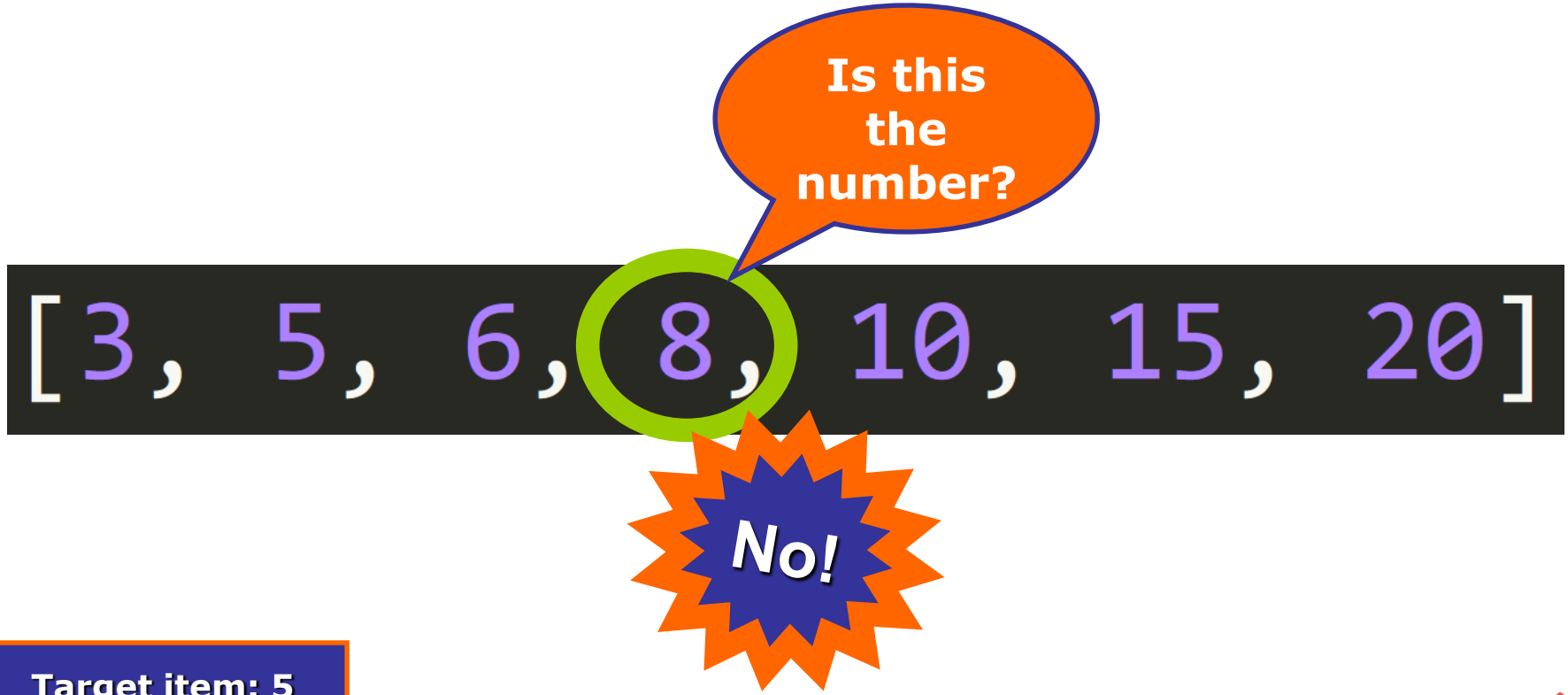
```
[3, 5, 6, 8, 10, 15, 20]
```

Target item: 5

Binary Search



Binary Search

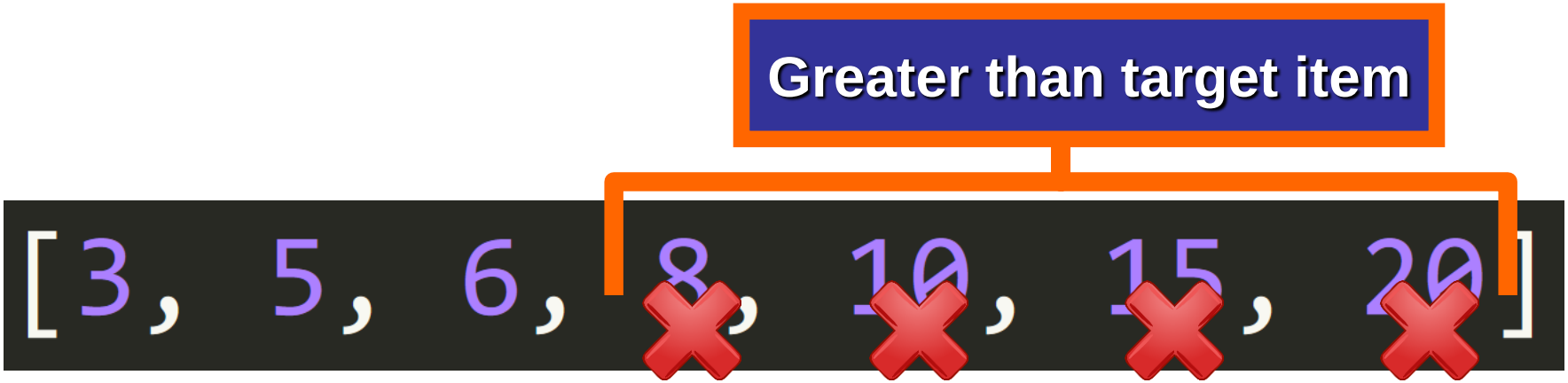


Binary Search

5 is smaller than 8

Target item: 5

Binary Search



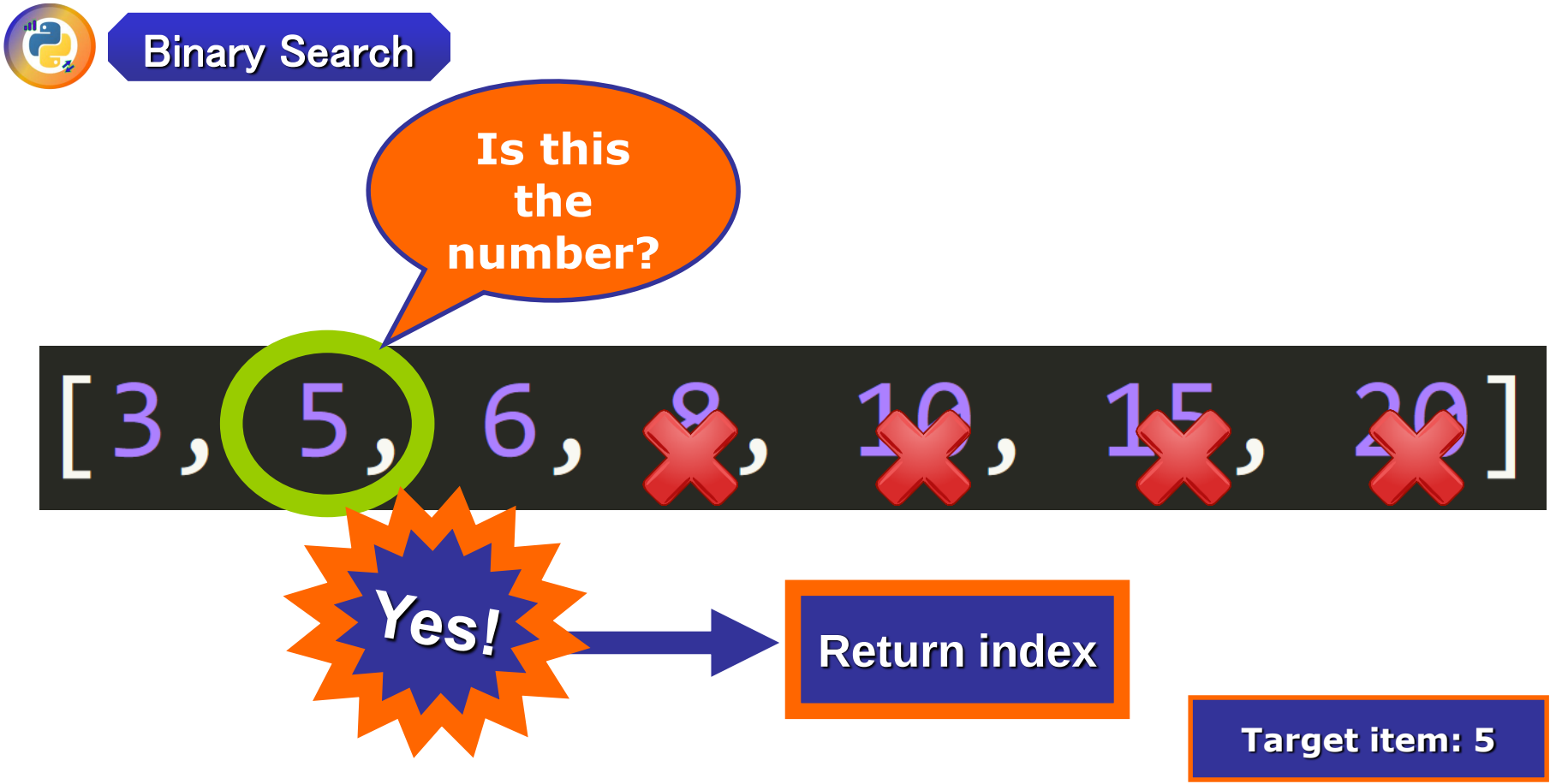
Target item: 5

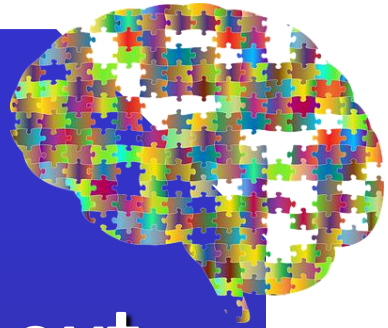
Binary Search

Target item could be here

Target item: 5

Binary Search


Binary Search
Is this the number?
[3, 5, 6, 8, 10, 15, 20]
Yes!
Return index
Target item: 5



Let's think about...
Big Picture

Binary Search

- **Start searching** in an interval that covers the list.
- Find the **middle** element. Check if it's the target element.
- If the target element is **greater** than the middle element, discard the lower half of the list.
- If it's **smaller**, discard the upper half.
- **Repeat** these steps for the new interval until the item is found or until all middle elements have been checked.
- **Return** the index if the item is found or -1 if it's not found.

Time to Code!