

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

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





Binary Search

7 Items

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

Target item: 15



Binary Search

7 Items

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

Middle element

Target item: 15



Binary Search

Is this
the
number?

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

No!

Target item: 15



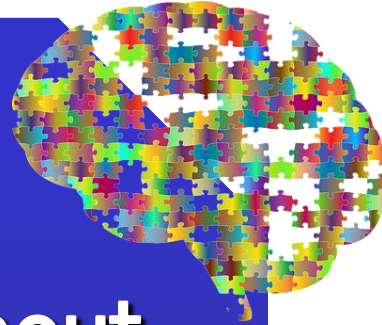
Binary Search

$15 > 8$

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

Target item: 15

**Let's think about...
where to search**





Binary Search

Smaller



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



Middle element



Binary Search

Smaller than target item

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

Diagram illustrating a binary search step. The array [3, 5, 6, 8, 10, 15, 20] is shown. The first four elements (3, 5, 6, 8) are crossed out with red X's, indicating they are smaller than the target item (15). An orange bracket groups these four elements, and a line connects this group to the text "Smaller than target item".

Target item: 15





Binary Search

The item could be here

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

The array is sorted. The first four elements (3, 5, 6, 8) are crossed out with red X's, indicating they are not the target. The last three elements (10, 15, 20) are highlighted with orange boxes, indicating they are the current search range.

Target item: 15





Binary Search

3 Items

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

Middle element

Target item: 15



Binary Search

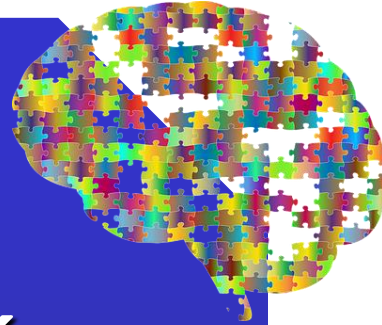
[~~3~~, ~~5~~, ~~6~~, ~~8~~, 10, 15, 20]

Is this
the
number?

Yes!

Return index

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





Binary Search

7 Items

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

Target item: 5





Binary Search

7 Items

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

Middle element

Target item: 5



Binary Search

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

Is this
the
number?

No!

Target item: 5



Binary Search

5 is smaller than 8

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

Target item: 5



Binary Search

Greater than target item

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

Diagram illustrating a binary search step. The array [3, 5, 6, 8, 10, 15, 20] is shown. The elements 8, 10, 15, and 20 are marked with red 'X' symbols, indicating they are greater than the target item (5). An orange bracket highlights the subarray [8, 10, 15, 20], which is the current search range.

Target item: 5





Binary Search

Target item could be here

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

The array [3, 5, 6, 8, 10, 15, 20] is shown. The first three elements (3, 5, 6) are enclosed in an orange box, indicating the current search range. The remaining elements (8, 10, 15, 20) are crossed out with red X's, indicating they are not in the current search range.

Target item: 5



Binary Search

3 Items

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

The array is shown with the first three elements (3, 5, 6) highlighted by an orange bracket. The elements 8, 10, 15, and 20 are each crossed out with a large red 'X'.

Middle element

Target item: 5



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!

