

Recursive binary search

```
def binary_search(array, target, low, high):
```

```
    if low > high:
```

```
        return -1
```

```
    mid = (low + high) // 2
```

```
    if array[mid] == target:
```

```
        return mid
```

```
    elif array[mid] < target:
```

```
        return binary_search(array, target, mid + 1, high)
```

```
    else:
```

```
        return binary_search(array, target, low, mid - 1)
```

The explanation of binary search:

- The algorithm starts by setting the low and high pointers to the first and last elements of the array, respectively.
- The algorithm then calculates the middle element of the array, which is the element at index $(low + high) // 2$.
- The algorithm compares the target value to the middle element.
 - If the target value is equal to the middle element, the algorithm returns its index.
 - If the target value is less than the middle element, the algorithm sets the high pointer to the middle element - 1 and continues the search in the lower half of the array.
 - If the target value is greater than the middle element, the algorithm sets the low pointer to the middle element + 1 and continues the search in the upper half of the array.
- The algorithm repeats steps 2-4 until the target value is found or the low and high pointers cross each other, in which case the algorithm returns -1.