

82. Binary search

PROGRAM:-

```
import time
```

```
def binary_search(arr, x):  
    left, right = 0, len(arr) - 1  
    while left <= right:  
        mid = left + (right - left) // 2  
        if arr[mid] == x:  
            return mid  
        elif arr[mid] < x:  
            left = mid + 1  
        else:  
            right = mid - 1  
    return -1
```

```
def find_binary_search_time(arr, x):  
    start_time = time.time() # Start time measurement  
  
    index = binary_search(arr, x) # Perform binary search  
  
    end_time = time.time() # End time measurement  
    elapsed_time = end_time - start_time  
  
    return index, elapsed_time
```

Example usage

```
example_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
search_element = 7
```

```
index, execution_time = find_binary_search_time(example_list, search_element)
```

```
if index != -1:
```

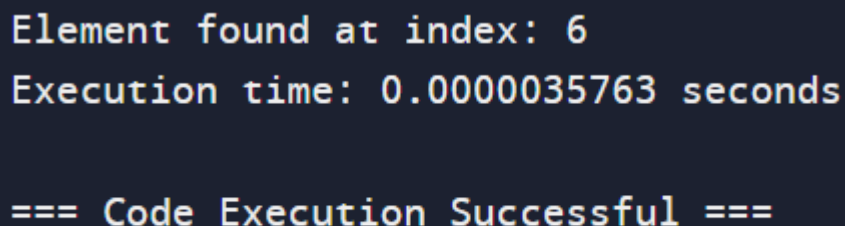
```
    print(f"Element found at index: {index}")
```

```
else:
```

```
    print("Element not found")
```

```
print(f"Execution time: {execution_time:.10f} seconds")
```

OUTPUT:-

A screenshot of a terminal window with a dark background and light-colored text. It shows the output of a binary search program. The first line says 'Element found at index: 6'. The second line says 'Execution time: 0.0000035763 seconds'. The third line is a separator '=== Code Execution Successful ==='.

```
Element found at index: 6  
Execution time: 0.0000035763 seconds  
  
=== Code Execution Successful ===
```

TIME COMPLEXITY:- $O(\log n)$