

1. 162. you are given a sorted array 3,9,14,19,25,31,42,47,53 and asked to find the position of the element 31 using Binary Search. Show the mid-point calculations and the steps involved in finding the element. Display, what would happen if the array was not sorted, how would this impact the performance and correctness of the Binary Search algorithm?

Input : N= 9, a[] = {3,9,14,19,25,31,42,47,53}, search key = 31

Output : 6

Code:

```
def binary_search(arr, low, high, key):
```

```
    steps = []
```

```
    while low <= high:
```

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

```
        steps.append(f"Low: {low}, High: {high}, Mid: {mid}, arr[Mid]: {arr[mid]}")
```

```
        if arr[mid] == key:
```

```
            steps.append(f"Element {key} found at index {mid}")
```

```
            return mid, steps
```

```
        elif arr[mid] < key:
```

```
            low = mid + 1
```

```
        else:
```

```
            high = mid - 1
```

```
    steps.append(f"Element {key} not found in the array.")
```

```
    return -1, steps
```

```
arr = [3, 9, 14, 19, 25, 31, 42, 47, 53]
```

```
key = 31
```

```
index, steps = binary_search(arr, 0, len(arr) - 1, key)
```

```
for step in steps:
```

```
    print(step
```

```
print(f"Index of element {key}: {index}")
```

output:

```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Desktop/csa0863_karthik/PROBLEM.py
Low: 0, High: 8, Mid: 4, arr[Mid]: 25
Low: 5, High: 8, Mid: 6, arr[Mid]: 42
Low: 5, High: 5, Mid: 5, arr[Mid]: 31
Element 31 found at index 5
Index of element 31: 5
PS C:\Users\karth>
```

Time complexity:  $f(n) = O(\log n)$