Pseudo-code: Binary Search with Random Target

Algorithm 1 BinarySearch

```
1: Input: A list of numbers.
2: Output: The target number found in the list.
3: Intuition:
     • Sort the list.
     • Create an index list from 0 to the length of the list minus one.
     • Select a random target index from the index list.
     • Perform binary search to find the target index.
4: Time Complexity: $O(\log n)$ for the binary search part, where $n$ is the number of elements in the list.
```

Algorithm 2 BinarySearch

```
1: function BinarySearch(list)
2:     Sort the list
3:     Create listIndex from 0 to len(list) - 1
4:     minIndex ← 0
5:     maxIndex ← len(list) - 1
6:     targetIndex ← random choice from listIndex
7:     guessIndex ← (minIndex + maxIndex) / 2
8:     return BinarySearchMid(minIndex, maxIndex, targetIndex, guessIndex, 0)
9: end function
```

Testing

- **Input:** {1, 3, 4, 7, 3, 2, 1, 0, 3, -5}
- **Output:** Depending on the random target, prints steps and final result of the binary search.

Algorithm 3 BinarySearchMid

```
function BinarySearchMid(minIndex, maxIndex, targetIndex, guessIndex, count)
    count ← count + 1
    if targetIndex = guessIndex then
        return list[targetIndex]
    else if list[targetIndex] > list[guessIndex] then
        minIndex ← guessIndex
        guessIndex ← (minIndex + maxIndex) / 2
        return BinarySearchMid(minIndex, maxIndex, targetIndex, guessIndex, count)
    else
        maxIndex ← guessIndex
        guessIndex ← (minIndex + maxIndex) / 2
        return BinarySearchMid(minIndex, maxIndex, targetIndex, guessIndex, count)
    end if
end function
```