Problem 2:

This algorithm includes modification of binary search and using a recursive function to find the rotation point. once the rotation point is found compare the target value with the 1st element of the array if > pass the left of rotation point or else pass the right array of the rotation point.

Time complexity:

Rotation_point() —> O(logn)

binary_search() —> O(logn)

binary_search_rotated() —> O(long + logn) === O(logn)

Space Complexity:

O(1)