recursiveSortSearch.java

```java
import java.util.Scanner;

public class recursiveSortSearch {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter size of array");
        int N = sc.nextInt();
        int[] arr = new int[N];
        System.out.println("Enter " + N + " numbers");
        for (int i = 0; i < N; i++) {
            arr[i] = sc.nextInt();
        }
        System.out.println("Original Array: ");
        print(arr);
        System.out.println("\nSorting using bubble sort");
        bubbleSort(arr, 0, 0);
        print(arr);
        System.out.println("\nEnter element to search for binary search");
        int e = sc.nextInt();
        sc.close();
        int search = binarySearch(arr, 0, N - 1, e);
        if (search == -1) {
            System.out.println("Element not found");
        } else {
            System.out.println("Element found at " + (search+1) + "th index");
        }
    }

    // Bubble Sort
    static void bubbleSort(int[] arr, int i, int j) {
        if (j == arr.length - i - 1) {
            if (i != arr.length - 2) {
                bubbleSort(arr, ++i, 0);
            } else return;
        } else {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
            bubbleSort(arr, i, ++j);
        }
    }

    // Binary Search
    static int binarySearch(int[] arr, int l, int u, int e) {
        if (u >= l) {
            int m = l + (u - l) / 2;
            if (arr[m] == e)
                return m;
            if (arr[m] > e)
                return binarySearch(arr, l, m - 1, e);
            return binarySearch(arr, m + 1, u, e);
        }
        return -1;
    }

    static void print(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}

```