1. Binary Search with Time

```
public class BS_Time {
    public static int search(int arr[], int key) {
        int low = 0;
        int high = arr.length - 1;

        while (low <= high) {
            int mid = low + (high - low) / 2;
            if (arr[mid] == key) {
                return mid;
            } else if (arr[mid] < key) {
                low = mid + 1;
            } else {
                high = mid - 1;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n;
        System.out.println("Enter size:");
        n = in.nextInt();
        int a[] = new int[n];
        System.out.println("Enter the elements in sorted manner:");
        for (int i = 0; i < n; i++) {
            a[i] = in.nextInt();
        }
        int key;
        System.out.println("Enter the search element:");
        key = in.nextInt();

        long start = System.nanoTime();
        int index = search(a, key);
        long stop = System.nanoTime();
        if (index != -1) {
            System.out.println("Element found at index " + index);
        } else {
            System.out.println("Element not found");
        }
        System.out.println("Time taken: " + (stop - start) + " nanoseconds");
    }
}
```

2. Quick Sort with Time

```
public class QuickSort {
    void quick(int arr[], int i, int j) {
        if (i < j) {
            int s = partition(arr, i, j);
            quick(arr, i, s - 1);
            quick(arr, s + 1, j);
        }
    }
    int partition(int arr[], int l, int h) {
        int p, i, j, temp;
        p = arr[l];
        i = l + 1;
        j = h;
        while (l < h) {
            while (arr[i] < p && i < h) {
                i++;
            }
            while (arr[j] > p) {
                j--;
            }
            if (i < j) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            } else {
                temp = arr[l];
                arr[l] = arr[j];
                arr[j] = temp;
                return j;
            }
        }
        return j;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of ele: ");
        int n = sc.nextInt();
        Random gen = new Random();
        int arr[] = new int[5000];
        for (int i = 0; i < n; i++) {
            arr[i] = gen.nextInt(1000);
        }
        System.out.println("Random ele: ");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
        long start = System.nanoTime();
        QuickSort qs = new QuickSort();
        qs.quick(arr, 0, n - 1);
        long stop = System.nanoTime();
        System.out.println("array after sorting: ");
        for (int i = 0; i < n; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
        System.out.println("Time taken: " + (stop - start));
    }
}
```