Code :

```
import java.util.*;

public class QuickSortAnalysis {

    // Deterministic Quick Sort (Pivot = Last Element)
    static void deterministicQuickSort(int[] arr, int low, int high) {
        if (low < high) {
            int p = partition(arr, low, high);
            deterministicQuickSort(arr, low, p - 1);
            deterministicQuickSort(arr, p + 1, high);
        }
    }

    // Standard partition function (Lomuto scheme)
    static int partition(int[] arr, int low, int high) {
        int pivot = arr[high]; // deterministic pivot
        int i = low - 1;

        for (int j = low; j < high; j++) {
            if (arr[j] <= pivot) {
                i++;
                swap(arr, i, j);
            }
        }
        swap(arr, i + 1, high);
        return i + 1;
    }
```

```
// Randomized Quick Sort
static void randomizedQuickSort(int[] arr, int low, int high) {
    if (low < high) {
        int p = randomizedPartition(arr, low, high);
        randomizedQuickSort(arr, low, p - 1);
        randomizedQuickSort(arr, p + 1, high);
    }
}

// Randomized partition
static int randomizedPartition(int[] arr, int low, int high) {
    int randomPivot = low + (int)(Math.random() * (high - low + 1));
    swap(arr, randomPivot, high); // swap random pivot with last element
    return partition(arr, low, high);
}

// Swap utility
static void swap(int[] arr, int i, int j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

// Main Function
public static void main(String[] args) {

    int[] arr1 = {30, 12, 18, 32, 13, 23, 89, 46, 10, 7};
    int[] arr2 = {30, 12, 18, 32, 13, 23, 89, 46, 10, 7};
```

```
        System.out.println("Unsorted Array : " + Arrays.toString(arr1));
        // Deterministic Quick Sort
        long start1 = System.nanoTime();
        deterministicQuickSort(arr1, 0, arr1.length - 1);
        long end1 = System.nanoTime();
        double time1 = (end1 - start1) / 1e6; // ms

        // Randomized Quick Sort
        long start2 = System.nanoTime();
        randomizedQuickSort(arr2, 0, arr2.length - 1);
        long end2 = System.nanoTime();
        double time2 = (end2 - start2) / 1e6; // ms

        System.out.println("\nSorted Array (Deterministic): " + Arrays.toString(arr1));
        System.out.println("Execution Time (Deterministic): " + time1 + " ms");

        System.out.println("\nSorted Array (Randomized): " + Arrays.toString(arr2));
        System.out.println("Execution Time (Randomized): " + time2 + " ms");
    }
}
```

Output :

```
Unsorted Array : [30, 12, 18, 32, 13, 23, 89, 46, 10, 7]

Sorted Array (Deterministic): [7, 10, 12, 13, 18, 23, 30, 32, 46, 89]
Execution Time (Deterministic): 0.0067 ms

Sorted Array (Randomized): [7, 10, 12, 13, 18, 23, 30, 32, 46, 89]
Execution Time (Randomized): 0.2286 ms
```