Code for Merge Sort:

import java.util.\*;

public class EXPT\_03\_MergeSort {

    static void Merge\_Arrays(int arr[], int start, int mid, int end) {

        int leftarr\_end = mid;

        int rightarr\_start = mid + 1;

        int l = start;

        while ((l <= leftarr\_end) && (rightarr\_start <= end)) {

            if (arr[start] < arr[rightarr\_start]) {

                start++;

            } else {

                int temp = arr[rightarr\_start];

                for (int j = rightarr\_start - 1; j >= start; j--) {

                    arr[j + 1] = arr[j];

                }

                arr[start] = temp;

                start++;

                leftarr\_end++;

                rightarr\_start++;

            }

        }

    }

    static void Merge\_Sort(int arr[], int start, int end) {

        if (start >= end) {

            return;

        }

        int mid = (start + end) / 2;

        Merge\_Sort(arr, start, mid);

        Merge\_Sort(arr, mid + 1, end);

        Merge\_Arrays(arr, start, mid, end);

    }

    public static void main(String args[]) {

        Scanner in = new Scanner(System.in);

        System.out.print("Enter the number of elements for array: ");

        int n = in.nextInt();

        int arr[] = new int[n];

        Random randomnumber = new Random();

        System.out.print("\nThe randomly-generated array is ");

        for (int i = 0; i < n; i++) {

            arr[i] = randomnumber.nextInt(1000);

            System.out.print("\t" + arr[i]);

        }

        long start = System.currentTimeMillis();

        Merge\_Sort(arr, 0, arr.length - 1);

        long end = System.currentTimeMillis();

        long time = end - start;

        System.out.print("\nAfter sorting by Merge Sort, the array elements are ");

        for (int i = 0; i < n; i++) {

            System.out.print("\t" + arr[i]);

        }

        System.out.print("\nThe number of elements in array are " + n + ".");

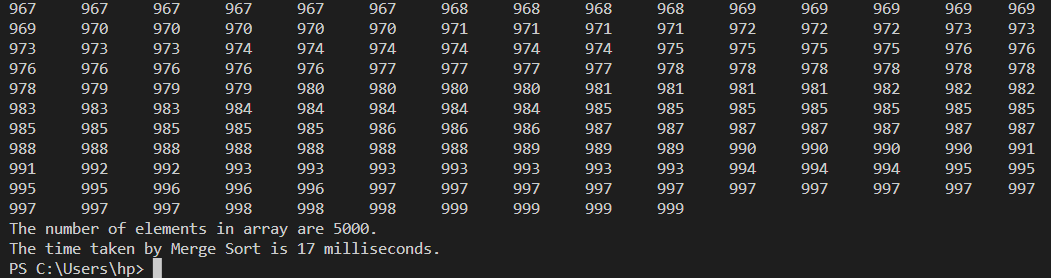
        System.out.print("\nThe time taken by Merge Sort is " + time + " milliseconds.");

    }

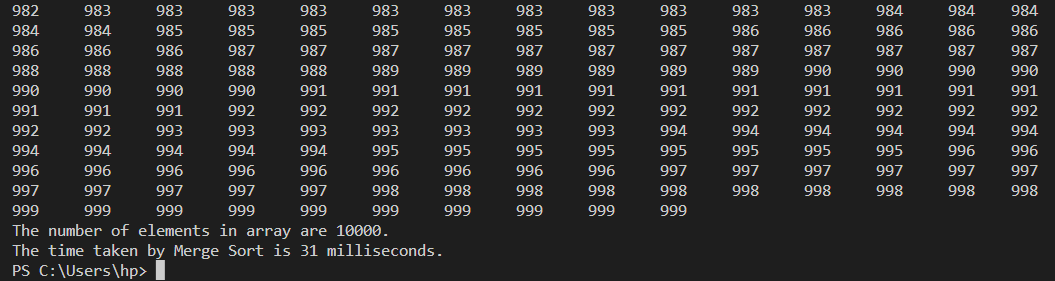
}

Case 1: When the number of elements is 5,000

17 milliseconds

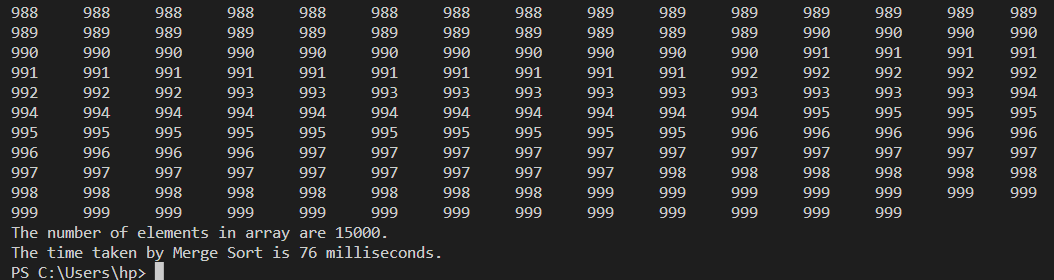


31 milliseconds



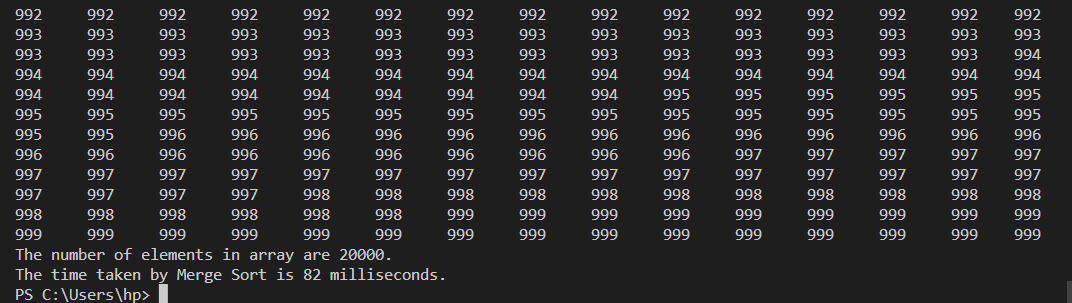
Case 3: When the number of elements is 15,000

76 milliseconds



Case 4: When the number of elements is 20,000

82 milliseconds



Case 5: When the number of elements is 25,000

126 milliseconds

