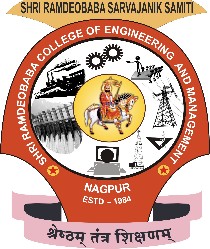
**Shri Ramdeobaba College of Engineering & Management Nagpur-13**

**Department of Computer Application**

**Session: 2023-2024**



**Submission for**

**Course Name:** Design Analysis and Algorithm Lab

**Course Code:** MCP546

**Name of the Student:** Jayesh Lalit Nandanwar

**Class Roll No:** 26

**Semester:** MCA II semester

**Shift:** 2

**Batch:** 2

Under the Guidance of

Prof. Manda Ukey

Date of submission: 06/03/2024

**Practical 4**

**Aim:** Perform insertion sort on the data sets that you have created in practical\_0.

Display the time taken to sort the elements from the files in ascending order. Consider random repeated and random unrepeated files.

Compare and show the time taken for selection sort.

**For file 1 (Sequential unrepeated numbers):**

**Code:** import java.io.\*;

import java.util.Scanner;

public class InsertionSortFile1 {

public static void main(String[] args) throws IOException {

Scanner scanner = new Scanner(System.in);

FileReader f = new FileReader("./sequentialUnrepeatedNumbers.txt");

Scanner fileScanner = new Scanner(f);

int[] array = new int[100001]; // maximum of 100000 elements in the file

int size = 0;

while (fileScanner.hasNextInt()) {

array[size++] = fileScanner.nextInt();

}

fileScanner.close();

long start = System.currentTimeMillis();

insertionSort(array);

long finish = System.currentTimeMillis();

long timeElapsed = finish - start;

System.out.println("\nTime taken for sorting: " + timeElapsed + " ms");

FileWriter writer = new FileWriter("./sequentialUnrepeatedNumbersSortedOutput.txt");

for (int i = 0; i < size; i++) {

writer.write(array[i]+"\n");

}

writer.close();

scanner.close();

}

public static void insertionSort(int arr[])

{

int n = arr.length;

for (int i = 1; i < n; ++i) {

int key = arr[i];

int j = i - 1;

while (j >= 0 && arr[j] > key) {

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

j = j - 1;

}

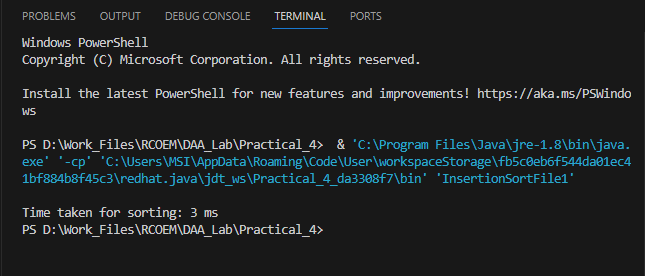
arr[j + 1] = key;

}

}

}

**Output:**



**Time Taken: 3 ms**

**For file 2 (Random unrepeated numbers):**

**Code:** import java.io.\*;

import java.util.Scanner;

public class InsertionSortFile2 {

public static void main(String[] args) throws IOException {

Scanner scanner = new Scanner(System.in);

FileReader f = new FileReader("./randomUnrepeatedNumbers.txt");

Scanner fileScanner = new Scanner(f);

int[] array = new int[100001]; // maximum of 100000 elements in the file

int size = 0;

while (fileScanner.hasNextInt()) {

array[size++] = fileScanner.nextInt();

}

fileScanner.close();

long start = System.currentTimeMillis();

insertionSort(array);

long finish = System.currentTimeMillis();

long timeElapsed = finish - start;

System.out.println("\nTime taken for sorting: " + timeElapsed + " ms");

FileWriter writer = new FileWriter("./randomUnrepeatedNumbersSortedOutput.txt");

for (int i = 0; i < size; i++) {

writer.write(array[i]+"\n");

}

writer.close();

scanner.close();

}

public static void insertionSort(int arr[])

{

int n = arr.length;

for (int i = 1; i < n; ++i) {

int key = arr[i];

int j = i - 1;

while (j >= 0 && arr[j] > key) {

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

j = j - 1;

}

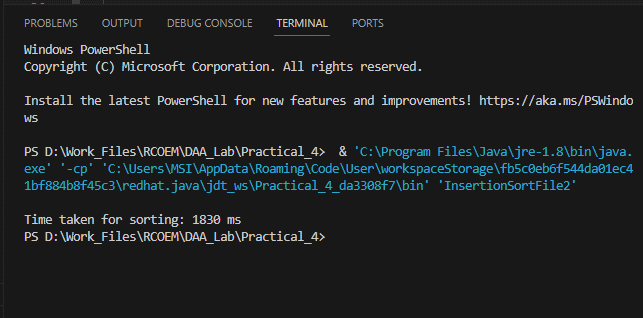
arr[j + 1] = key;

}

}

}

**Output:**



**Time Taken: 1830 ms**

**For file 3 (Random repeated numbers):**

**Code:** import java.io.\*;

import java.util.Scanner;

public class InsertionSortFile3 {

public static void main(String[] args) throws IOException {

Scanner scanner = new Scanner(System.in);

FileReader f = new FileReader("./randomRepeatedNumbers.txt");

Scanner fileScanner = new Scanner(f);

int[] array = new int[100001]; // maximum of 100000 elements in the file

int size = 0;

while (fileScanner.hasNextInt()) {

array[size++] = fileScanner.nextInt();

}

fileScanner.close();

long start = System.currentTimeMillis();

insertionSort(array);

long finish = System.currentTimeMillis();

long timeElapsed = finish - start;

System.out.println("\nTime taken for sorting: " + timeElapsed + " ms");

FileWriter writer = new FileWriter("./randomRepeatedNumbersSortedOutput.txt");

for (int i = 0; i < size; i++) {

writer.write(array[i]+"\n");

}

writer.close();

scanner.close();

}

public static void insertionSort(int arr[])

{

int n = arr.length;

for (int i = 1; i < n; ++i) {

int key = arr[i];

int j = i - 1;

while (j >= 0 && arr[j] > key) {

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

j = j - 1;

}

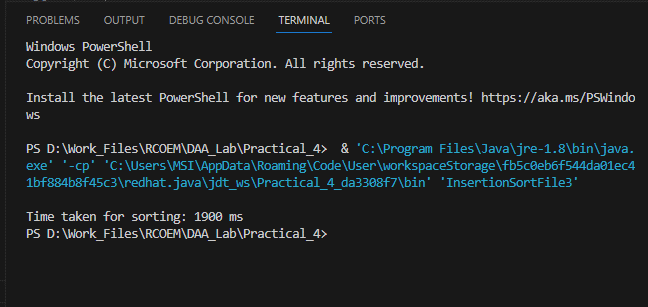
arr[j + 1] = key;

}

}

}

**Output:**



**Time Taken: 1900 ms**

**Comparison between time taken:**

|  |  |  |
| --- | --- | --- |
| **Sorting Algorithm**  Insertion Sort  Selection Sort | **File Data Type** | **Time Taken**  **(in ms)** |
| Selection Sort | Sequential Unrepeated Numbers | 1103 |
| Random Unrepeated Numbers | 1591 |
| Random Repeated Numbers | 1690 |
| Insertion Sort | Sequential Unrepeated Numbers | 3 |
| Random Unrepeated Numbers | 1830 |
| Random Repeated Numbers | 1900 |

**Observations:**

* Time taken to sort Random Unrepeated Numbers and Random Repeated Numbers is more in Insertion Sort (1830 ms and 1900 ms respectively) than Selection Sort (1591 ms and 1690 ms respectively)
* Time taken to sort Sequential Unrepeated Numbers is less in Insertion Sort (3 ms) than Selection Sort (1103 ms)