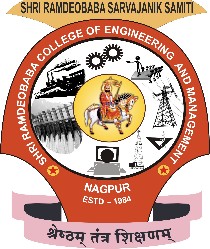
**Shri Ramdeobaba College of Engineering & Management Nagpur-13**

**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: 20/02/2024

**Practical 2**

**Aim:** Perform selection sort on all the three data sets which you have created in practical\_0.

Display the time taken to sort the elements from the files in ascending order. Consider all the three files.

Compare the time taken for all these cases.

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

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

import java.io.IOException;

import java.util.Scanner;

public class SelectionSortFile1 {

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();

selectionSort(array, size);

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 selectionSort(int[] arr, int n) {

for (int i = 0; i < n - 1; i++) {

int minIndex = i;

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

if (arr[j] < arr[minIndex]) {

minIndex = j;

}

}

int temp = arr[minIndex];

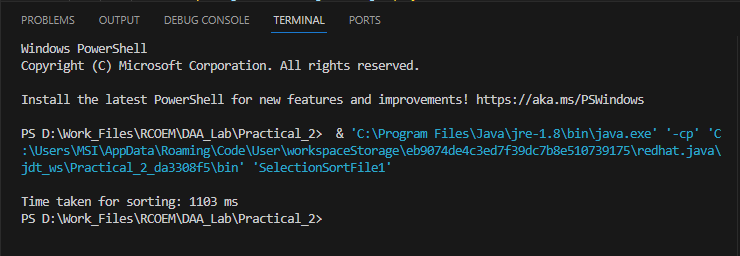
arr[minIndex] = arr[i];

arr[i] = temp;

}

}

**Output:**



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

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

import java.io.IOException;

import java.util.Scanner;

public class SelectionSortFile2 {

    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();

        selectionSort(array, size);

        long finish = System.currentTimeMillis();

        long timeElapsed = finish - start;

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

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

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

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

        }

        writer.close();

        scanner.close();

    }

    public static void selectionSort(int[] arr, int n) {

        for (int i = 0; i < n - 1; i++) {

            int minIndex = i;

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

                if (arr[j] < arr[minIndex]) {

                    minIndex = j;

                }

            }

            int temp = arr[minIndex];

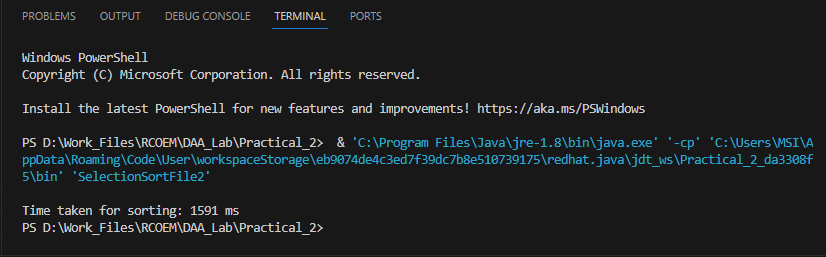
            arr[minIndex] = arr[i];

            arr[i] = temp;

        }

    }

**Output:**



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

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

import java.io.IOException;

import java.util.Scanner;

public class SelectionSortFile3 {

    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();

        selectionSort(array, size);

        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 selectionSort(int[] arr, int n) {

        for (int i = 0; i < n - 1; i++) {

            int minIndex = i;

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

                if (arr[j] < arr[minIndex]) {

                    minIndex = j;

                }

            }

            int temp = arr[minIndex];

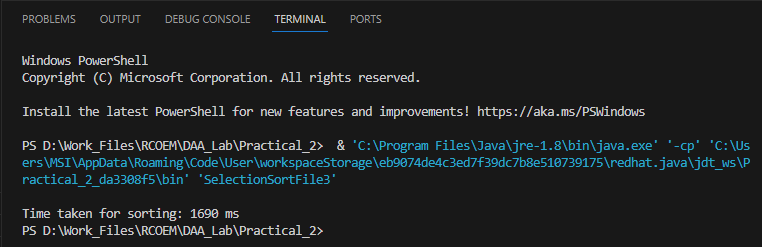
            arr[minIndex] = arr[i];

            arr[i] = temp;

        }

    }

**Output:**

****

**Comparison between time taken:**

|  |  |  |  |
| --- | --- | --- | --- |
| **File** | **File 1** | **File 2** | **File 3** |
| **Data Type** | Sequential Unrepeated Numbers | Random Unrepeated Numbers | Random Repeated Numbers |
| **Time Taken (ms)** | 1103 | 1591 | 1690 |

**Observations:**

* Time taken to sort Random Repeated Numbers is maximum (1690 ms)
* Time taken to sort Sequential Unrepeated Numbers is minimum (1103 ms)
* Time taken for :

Sequential Unrepeated Numbers < Random Unrepeated Numbers < Random Repeated Numbers