**Lab Report 2**

**CS303 Algorithms and Data Structures**

**Sam Lazrak**

**Objectives:**

* Implement insertion sort algorithm
* Evaluate performance of insertion sort with increasing array size

**In-class Assignment:**

1. Implement a method that will sort a given array using the insertion sort algorithm
2. Write a driver program to test the insertion algorithms implemented in question 1. Read the input file “input\_100.txt” for the input numbers and store them in an array. Sort this array using insertion sort.

The following source code was written:

package lab2;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Arrays;

import java.util.Scanner;

public class InsertionSort {

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

File file = new File("input\_100.txt");

Scanner s = new Scanner(file);

int i = 0;

while(s.hasNextInt()) {

array[i] = s.nextInt();

i++;

}

insertionSort(array);

System.out.println(Arrays.toString(array));

}

public static int[] insertionSort(int[] a) {

for (int j = 1; j < a.length; j++) {

int key = a[j];

int i = j - 1;

while(i >= 0 && a[i] > key) {

a[i+1]= a[i];

i = i - 1;

}

a[i+1] = key;

}

return a; } }

Output:

[0, 0, 0, 1, 1, 3, 4, 5, 7, 8, 10, 12, 12, 13, 14, 15, 15, 16, 18, 22, 22, 22, 23, 24, 24, 25, 25, 27, 29, 29, 30, 34, 36, 36, 37, 38, 38, 39, 40, 40, 41, 43, 45, 47, 47, 49, 50, 50, 52, 52, 53, 54, 54, 54, 54, 57, 60, 61, 63, 64, 64, 67, 67, 68, 69, 69, 70, 72, 73, 73, 74, 74, 75, 75, 75, 76, 77, 77, 78, 80, 81, 81, 81, 83, 83, 84, 85, 85, 86, 87, 88, 89, 92, 93, 94, 94, 94, 95, 96, 98]

**Homework Assignment:**

1. Test the program for the different size input files provided in Canvas.

2. Record the runtime for insertion sort on various sized arrays by using the provided files. Comment on how the execution time of insertion sort varies with size of the input array. Use a table or plot to summarize the results and document your observations and explanations in the report.

Source code:

package lab2;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Arrays;

import java.util.Scanner;

public class InsertionSort {

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

File[] files = new File[7];

files[0]= new File("input\_100.txt");

files[1]= new File("input\_1000.txt");

files[2]= new File("input\_5000.txt");

files[3]= new File("input\_10000.txt");

files[4]= new File("input\_50000.txt");

files[5]= new File("input\_100000.txt");

files[6]= new File("input\_500000.txt");

int[] size = {100, 1000, 5000, 10000, 50000, 100000, 500000};

for(int i = 0; i <= 6; i++){

int length = size[i];

int[] array = new int[length];

Scanner s = new Scanner(files[i]);

int j = 0;

while(s.hasNextInt()) {

array[j] = s.nextInt();

j++; }

long sortTime = System.nanoTime();

insertionSort(array);

sortTime = System.nanoTime() - sortTime;

System.out.println(Arrays.toString(array));

System.out.println("Array Size: " + size[i] + "\n" + "Sort Time: " + sortTime + "\n");

}

}

public static int[] insertionSort(int[] a) {

for (int j = 1; j < a.length; j++) {

int key = a[j];

int i = j - 1;

while(i >= 0 && a[i] > key) {

a[i+1]= a[i];

i = i - 1;

}

a[i+1] = key;

}

return a; } }

Output:

|  |  |
| --- | --- |
| Array Size: 100 | Sort Time: 86764 |
| Array Size: 1000 | Sort Time: 10709819 |
| Array Size: 5000 | Sort Time: 41013923 |
| Array Size: 10000 | Sort Time: 38548084 |
| Array Size: 50000 | Sort Time: 610990571 |
| Array Size: 100000 | Sort Time: 1052030287 |
| Array Size: 500000 | Sort Time: 39185028101 |

Each array was sorted and did so correctly, the output was too long to input into this report. Insertion sort is O(n2) and it is noticeable as the array sizes increase.