**Assignment #1 (5%)**

Submission deadline: Thursday, **January 27, 2022** **(14:00)**

**Name: Talha Abdullah Punjabi QUID: 201903446**

**Important Notes (must read):**

1. When submitting your work, you must use Blackboard, **NO other means like email submissions, etc., are accepted.**
2. Assignments are to be solved **individually**. It is expected that you may discuss the problems with your classmates, however, you should **NOT** discuss the solutions NOR write-up solutions together NOR copy-and-paste solutions that are not YOURS.
3. A mark of zero (0) will be awarded for the whole assessment in which the plagiarism was found to occur. Even if a single question is plagiarized**, the whole assignment will get zero** (0).
4. Submit your work as instructed below ***before*** the deadline**. No extension will be provided.**

If you have any uncertainty, questions or doubts about any of the above, ask Eng. Alaa Hussein [alaa.hussein@qu.edu.qa](mailto:alaa.hussein@qu.edu.qa) . There are five questions in this assignment, each of them is 20 points.

**Q1. (Section 1.1.)** Write a **Java program** that finds the **largest number** in a list (an array) of *n* numbers.

Answer:

**package** Assignment1;

**import** java.util.Scanner;

**public** **class** LargeNumberArray {

**public** **static** **void** main(String []args) {

Scanner writer=**new** Scanner(System.***in***);

System.***out***.print("Enter the Size of the Array : ");

**int** size=writer.nextInt();

System.***out***.println("Size of the array is : "+size);

**int**[] arr=**new** **int**[size];

System.***out***.println("Enter Array Elements :");

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

System.***out***.print(i+"th index : ");

**int** n=writer.nextInt();

arr[i]=n;

}

**int** max=arr[0];

**for**(**int** i=1;i<size;i++) {

**if**(max<arr[i]) {

max=arr[i];

}

}

System.***out***.println("Max Number in Array is : "+max);

}}

Output:

Graphical user interface, text, application

Description automatically generated

**Q2. (Section 1.1.)** Write a **Java program** that finds the ***m* smallest numbers** in a list of *n* numbers.

Answer:

**package** Assignment1;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** smallNumbers {

**public** **static** **void** main(String [] args) {

Scanner write=**new** Scanner(System.***in***);

System.***out***.print("Enter the Size of the List : ");

**int** n=write.nextInt();

System.***out***.println("Enter the List values : ");

ArrayList<Integer> array=**new** ArrayList<>();

ArrayList<Integer> array2=**new** ArrayList<>();

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

System.***out***.print(i+"th index = ");

**int** k=write.nextInt();

array.add(k);

}

System.***out***.print("Enter the number of smallest numbers :");

**int** smalls=write.nextInt();

**int** index=0;

**for**(**int** i=0;i<smalls;i++) {

**int** min=array.get(0);

**for**(**int** j=0;j<array.size();j++) {

**if** (min>array.get(j)) {

min=array.get(j);

index=j;

}

}

array.remove(index);

array2.add(min);

}

System.***out***.println("Smallest Numbers in the list are: ");

**for**(**int** i=0;i<smalls;i++) {

System.***out***.print(array2.get(i)+" ");

}}}

Output:

Graphical user interface, text, email

Description automatically generated

**Q3. (Section 1.2.)** Under what circumstances, when a searching operation is needed, would Sequential Search (Algorithm 1.1.) not be appropriate?

Answer: Sequential search would have to go over the whole list of elements. Sequential search would not be appropriate when to find instance of y, then the algorithm must check for all the elements inside the list. In case, the required y is at the end of list or not there at all, it will consume a lot of time and is not efficient.

**Q4. (Section 1.3)** There are two algorithms called Alg1 and Alg2 for a problem of size n. Alg1 runs in n2 microseconds and Alg2 runs in 100n logn microseconds. Alg1 can be implemented using 4 hours of programmer time and needs 2 minutes of CPU time. On the other hand, Alg2 requires 15 hours of programmer time and 6 minutes of CPU time. If programmers are paid 20 dollars per hour and CPU time costs 50 dollars per minute, how many times must a problem instance of size 500 be solved using Alg2 in order to justify its development cost?

Answer:

Payment per hour = 20 $

Alg1 payment = 20 x 4 = 80 $

Alg2 payment = 15 x 20 = 300 $

Must be problem of instance of size 500

Runtime of Alg1= n2 microseconds = (500)2 =250,000 microseconds

Runtime of Alg2=100n logn= 100 x 500 x log(500) = 50,000 x log(500) = 134,998 microseconds

Runtime cost of alg1=250,000 x 50$/min

Runtime cost of alg2=134,998 x 50$/min

For Alg1 = 80+ (Runtime cost of Alg1)x

For Alg2= 300+ (Runtime cost of Alg2)x

Both are equal as an instance of x,

80 +(250,000 x 50$/min) = 300 +(134,998 x 50$/min)

Solving for x,

x=2293.87

Must run almost 2294 times

**Q5. (Section 1.4.) Write the complexity categories of the following functions:**

1. ****

Answer: (please write your answer here, add required space if needed)

**Ans)** Θ (n2)

1. ****

Answer: (please write your answer here, add required space if needed)

**Ans)** θ (n)

1. ****

Answer: (please write your answer here, add required space if needed)

**Ans)** θ (2n)