GTU Department of Computer Engineering

CSE 222/505 - Spring 2023

Homework 2

Due date: March 26, 2023 – 23:59

1) ***(30 pts)*** For each of the function pairs below, show whether $f(n) = O(g(n))$, or$f(n) = \Omega(g(n))$, or $f(n) = \Theta(g(n))$ by using the limit approach.

   a) $f(n) = n^2 + 7n \ and \ g(n) = n^3 + 7$

   b) $f(n) = 12n + \log_2 \quad n^2 \ and \ g(n) = n^2 + 6n$

   c) $f(n) = n \cdot \log_2 \quad 3n \ and \ g(n) = n + \log_2(8 \cdot n^3)$

   d) $f(n) = n^n + 5n \ and \ g(n) = 3 \cdot 2^n$

   e) $f(n) = \sqrt[3]{2n} \ and \ g(n) = \sqrt{3n}$

2) ***(20 pts)*** Analyze the *worst-case* time complexity of the following methods.

   PS: For each method, if there is an array, assume its length as $n \ where \ n \in Z^+$.

   a)

```
static void methodA (String names[]) {
    for (int i = 0; i < names.length ; i++)
       System.out.println(names[i]);
  }
```

   b)

```
static void methodB () {
    String[] myArray = new String[] {"CSE222",
"CSE505", "HW2"};
    for (int i = 0; i < myArray.length; i++)
       methodA(myArray);
  }
```

c)

```
static void methodC (int numbers[]) {
    int i = 0;
    while (i < numbers.length)
       System.out.println(numbers[i]);
  }
```

d)

```
static void methodD (int numbers[]) {
    int i = 0;
    while (numbers[i] < 4)
       System.out.println(numbers[i++]);
  }
```

3) ***(20 pts)*** What is the difference between the time complexities of the following methods? Which one is more advantageous?

```
  static void withoutLoop(int [] myArray) {
     int i = 0;
     System.out.println(myArray[i++]);
     System.out.println(myArray[i++]);
     System.out.println(myArray[i++]);
     System.out.println(myArray[i++]);
     System.out.println(myArray[i++]);
     /*
     ...
     ...
     assume that the 'System.out.println' is called
myArray.length times in total
     */
     System.out.println(myArray[i++]);
  }
```

```
  static void witLoop(int [] myArray) {
     for (int i = 0; i < myArray.length; i++)
        System.out.println(myArray[i]);
  }
```

4) ***(10 pts)*** Consider an array of n integers ($n \in Z^{+}$). You do not have any information on whether the array is sorted or not, and you are supposed to check if the array contains a specific integer. Considering all possible inputs, can you solve this problem in constant time? If so, write down the pseudo-code of the algorithm and analyze its time complexity. If not, explain why.

5) ***(20 pts)*** Consider two integer arrays *A* and *B as follows:*

$$A = [a_0,\ a_1, \dots, a_{n-1}]$$
$$B = [b_0,\ b_1, \dots, b_{m-1}]$$

where $n,\ m \in\ Z^{+}$. Design *a linear time algorithm* to find the minimum value of $a_i \cdot b_j$ where $0 \le i < n$ and $0 \le j < m$. Explain your algorithm (along with the pseudo-code) and analyze its worst-case time complexity.

GENERAL RULES

- No late submissions are accepted.
- You should upload a pdf file of your **handwritten** work.
- Any immediate answer without justification **will not be graded.**
- If **any part** of your work is detected as a cheat, **you will get -100.**