CSE222 / BİL505
Data Structures and Algorithms
Homework #2
Due Date: 18/03/2024, 23:59

Q1. ***(35 pts)*** For each of the function pairs below, show whether $f(n) = O(g(n))$, or $f(n) = \Omega(g(n))$, or $f(n) = \Omega(g(n))$ by using the limit approach. Make sure you solve ***all*** indeterminities and show the details of your work.

a) $f(n) = (n^2 - 3n)^2 \ and \ g(n) = 5n^3 + n$
b) $f(n) = n^3 \ and \ g(n) = \log_2 \ n^4$
c) $f(n) = 5n \cdot \log_2(4n) \ and \ g(n) = n \cdot \log_2(5^n)$
d) $f(n) = n^n \ and \ g(n) = 10^n$
e) $f(n) = 8n \cdot \sqrt[5]{2n} \ and \ g(n) = n \cdot \sqrt[3]{n}$

Limit approach:

- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \rightarrow f(n) = O(g(n))$
- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \rightarrow f(n) = \Omega(g(n))$
- $0 < \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty \rightarrow f(n) = \Theta(g(n))$

Q2. ***(35 pts)*** Analyze the ***worst-case*** time complexity of the following methods.

PS: Assume that all of the arrays has a length of $n, where \ n \in Z^+$.

a)

```
static void methodA (String str_array[]) {
    for (int i = 0; i < str_array.length; i++)
      str_array [i] = "";
}
```

b)

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

c)

```
static void methodC (String str_array[]) {
    for (int i = 0; i < str_array.length; i++)
        for (int j = 0; j < str_array.length; j++)
            methodB (str_array);
}
```

d)

```
static void methodD (String str_array[]) {
    for (int i = 0; i < str_array.length; i++){
        System.out.println(str_array[i]);
        str_array[i--] = "";
    }
}
```

e)

```
static void methodE (String str_array[]) {
    for (int i = 0; i < str_array.length; i++)
      if (str_array [i] == "")
          break;
}
```

Q3. ***(30 pts)*** Design an algorithm to find the maximum difference between two elements of a given array $A = [a_0, a_1, \dots, a_{n-1}]\ where\ a_i \in \mathbb{Z}\ for\ i \in \mathbb{N}\ and\ i < n$. Provide the ***pseudo-code*** of the algorithm along with an explanation and analyze its worst-case time complexity. Repeat this process for the following cases:

a) Assuming the array is sorted in ascending order.
b) Assuming the array is not sorted.

PS: To be graded, your algorithms' worst-case time complexities should be linear time at most.

General Information About the Homework:

- Cheating is not permitted. The students who cheat will receive NA from the course.
- You should provide your ***handwritten*** work. Otherwise it will not be accepted. You can either hand it over in person, or you can upload the pictures of your work as ***a single PDF file*** to Teams. If you are going to hand it in person, please find me in my office (1st floor, 122) during working hours.
- The Problem Session on March 12 will describe the homework in detail. If you have any questions, you should ask them in PS. Any other questions before/after the PS will be ignored unless there is a mistake or missing information in this PDF. In such a case, the announcement will be made on Teams, you are responsible for reading them in time.
- Late submission is not allowed. The due date will not be postponed.