CSE 321 – Fall 2023

Homework 1

Due Date: (2/11/2023) 23:55

1. In each of the following situations, determine whether $f \in O(g)$, $f \in \Omega(g)$, or both (in which case $f \in \Theta(g)$). Provide explicit explanations for your answers. For at least half of the examples, perform a limit analysis.

	f(n)	g(n)
a)	2^n	2^{2n}
b)	n^2	n^3
c)	$3n + 1$	$2n - 5$
d)	$4n^2$	n^2
e)	$log2(n)$	$log10(n)$
f)	2^n	3^n
g)	n^3	$1000n^2$
h)	$5n + 4$	$2n + 2$
i)	$\sqrt{n}$	$log2(n)$
j)	2^n	2^{n+1}

2. List the following functions in order of their growth and provide proof for your claims.

$$\frac{1}{2n}, log(n), \sqrt{n + 5}, n + 1, 10^n, n^2 \log(n), 2^n, n!, \ n^{2^n}$$

3. Provide pseudo code for the following operations on a given binary search tree (BST) with a height of n. Analyze the time complexity (in terms of Big-Oh notation) of your code for each of the following:

   a) Merging with another BST of height n.
   b) Finding the kth smallest element in the BST.
   c) Balancing the BST.
   d) Finding elements within a specified value range.

4. Calculate the time complexity (in terms of <mark>big</mark> -Oh notation) of the following program.

```
i = 2
while i <= n:
  if i % 2 != 0:
    i = i - 1
  else:
    i = i * i
    i = i + 1
  print(i)
```

5. Suppose you have an array of n elements, where each element can be either even or odd with a probability distribution of 20% even and 80% odd. Propose an algorithm that identifies the first even element in the array. Describe the algorithm and analyze its average-case time complexity.

Notes:

- Your answer must be handwritten and submitted via the Course MS Teams page.
- Pseudocodes should be submitted as actual Python code and submitted as separate files together with your handwritten solutions.
- If you have any questions, you can send an email to b.koca@gtu.edu.tr
- Please complete your homework individually; group studies will be regarded as cheating.