Practice Questions
CSE 2046 Analysis of Algorithms, Spring 2021

1. Solve the following recurrence relations using **backward substitution method**:
(a) T(n)=T(n-2)+n for n>1, T(1)=1, T(2)=2. for n>1, T(1)=1, T(2)=2. *(Solve for both odd and even values of n.)*

(b) T(n)=4T($\lceil n/2 \rceil$)+n for n>1, T(1)=1. *(solve for* $n = 2^k$*).*

(c) T($\sqrt{n}$)+1, T(2) = 1. *(solve for* $n = 2^{2^k}$*).*

(d) G(n) = 2G(n-1) if n is odd and n≥1;
G(n) = G(n-1) + G(n-2) if n is even and n>0;
G(0) = 1; *(Find the solution for both odd and even values of n.)*

2. (n-1)!, 2nlog$(2n+5)^5$, 3^{n-1}, $115n^{10}$+$n3^n$ +1, $\ln^2(\sqrt{n})$, $\sqrt{n + 5\log_3 n}$, $\log_{10}(3^{\sqrt{n}}) - 5$
(a) For each of the above functions, indicate the class Θ(g(n)) the function belongs to. (Use the simplest g(n) possible in your answers.)

(b) List the above functions according to their order of growth from the lowest to the highest.

3. How many ones does the following procedure print when run with input n? Compute the best bounds you can: the exact value if possible, a big-Θ expression if you can't find the exact value, or big-O and big-Ω bounds if you can't find a big-Θ expression.

```
Ones(n):
   if n = 0:
         print 1
   else:
         for i = 1 to 2^n:
               Ones(n-1)
```

4. How many lines, as a function of n (in Θ(·) form), does the following programs print? Write a recurrence relation and solve it.

(a)

```
function func1(n)

if n = 1:
      print_line("Ayinesi iştir kişinin lafa bakılmaz.")
else:
      func1(⌊n/3⌋)
      for i=1: ⌊n/3⌋
            print_line("Ayinesi iştir kişinin lafa bakılmaz.")
      end for
```

(b)

```
function func2($n$)

if $n$ > 1:
      func2($\lfloor n/3 \rfloor$)
      print_line("Görünür kişinin rütbe-i aklı eserinde.")
      func2($\lfloor n/3 \rfloor$)
      print_line("Görünür kişinin rütbe-i aklı eserinde.")
      func2($\lfloor n/3 \rfloor$)
```

5. Consider the following program.

```
void func1(int n, int x) {
if (x <= 0)
      foo1 (n);
else
      foo2 (n);
```

It is known that x can get both negative and positive values (but not certainly with the same probability). Time complexities of **foo1(n)** and **foo2(n)** are given in the following table.

	Worst case	Best case	Average case
foo1 (n)	$\Theta(n^2)$	$\Theta(n)$	$\Theta(n^2)$
foo2 (n)	$\Theta(n \log n)$	$\Theta(1)$	$\Theta(n \log n)$

For each of the following, indicate whether it is “true”, “false”, or “there is no enough information”. Give a short reasoning. Answers without any comments will not be graded.

(a) Time complexity of func1 is in $\Theta(n^2)$
(b) Worst case time complexity of func1 is in $\Omega(n^2)$
(c) Average case time complexity of func1 is in $O(n^2)$
(d) Average case time complexity of func1 is in $\Theta(n^2)$

6. What is the time complexity of the following function? Indicate your answer in **Θ(·)** form.

```
void func2(int n) {
 int i = n;
      int x = 0;
      int count = 0;

      while (i > 1){
            x = x + 2;
            i = i/3;
      }
      for(int j = 1; j <= x; j++)
            for(int k = 1; k <= x; k++)
                  count = count + 1;
}
```