COL202: Discrete Mathematical Structures
Tutorial/Homework: 06

1. Discuss Quiz-04 in case required.
2. Answer the following:
   (a) State true or false: $2^{\sqrt{\log_2 n}}$ is $O(n)$.
   (b) Give reason for your answer to part (a).
3. Answer the following:
   (a) State true or false: 3^n is $O(2^n)$.
   (b) Give reason for your answer to part (a).
4. Consider functions $f(n) = 10n2^n + 3^n$ and $g(n) = n3^n$. Answer the following:
   (a) State true or false: $f(n)$ is $O(g(n))$.
   (b) State true or false: $f(n)$ is $\Omega(g(n))$.
   (c) Give reason for your answer to part (b).
5. Show using induction that for all $n \geq 0$, $1 + \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + ... + \frac{1}{2^n} = \frac{1-(\frac{1}{2})^{n+1}}{1-\frac{1}{2}}$.
6. Consider the following recursive function:

```
F(n)
  - If (n > 1) F(n/2)
  - Print("Hello World")
```

   Let $R(n)$ denote the number of times this function prints "Hello World" given the positive integer n as input.
   (a) What is $R(n)$, in big-O notation as a function of n?
   (b) Give reason for your answer to part (a).
7. Consider the following recursive algorithm that is supposed to convert any positive integer in decimal to binary format. $\lfloor . \rfloor$ denotes the floor function, $n\%2$ denotes the remainder when n is divided by 2, and $||$ denotes concatenation.

```
RecDecimalToBinary(n)
  - if(n = 0 or n = 1)return(n)
  -return(RecDecimalToBinary(⌊n/2⌋) || n%2)
```

Prove that the above algorithm is correct.

8. Show that:

   (a) If $d(n) = O(f(n))$ and $f(n) = O(g(n))$, then $d(n) = O(g(n))$.
   (b) $\max\{f(n), g(n)\} = O(f(n) + g(n))$.
   (c) If $a(n) = O(f(n))$ and $b(n) = O(g(n))$, then $a(n) + b(n) = O(f(n) + g(n))$.

9. Consider the two algorithms given below. In the input, A denotes an integer array and n denotes the size of the array. Analyse the running time of these algorithms and express the running time in big-O notation.

```
Alg1(A, n)
  - for i = 1 to n
    - j ← i
    - while(j < n)
      - A[j] ← A[j] + 10
      - j ← j + 3
```

```
Alg2(A, n)
  - for i = 1 to n
    - for j = 2i to n
      - A[i] ← A[j] + 1
```

10. Consider the following problem:

    ALL-ZEROS: Given the description $\langle A \rangle$ of an algorithm A, determine if this algorithm halts on all inputs with output 0.

    An algorithm P is said to solve the above problem if $P(\langle A \rangle)$ halts and outputs 1 when A is an algorithm that halts on all inputs producing 0, and it halts and outputs 0 otherwise. Does there exist an algorithm P that solves the problem ALL-ZEROS?