MAL 609 Basic Computer Science
Tutorial Sheet -I

(**Asymptotic)**

1. Suppose $T_1(n) = \Omega(f(n))$ and $T_2(n)= \Omega(g(n))$. Which of the following statements are true?
   (a) $T_1(n)+T_2(n)= \Omega(\max \{f(n), g(n)\})$.
   (b) $T_1(n)T_2(n)= \Omega(f(n)g(n))$.
   (c)
2. If f(n) is a k^{th} degree polynomial, then show that
   (a) $f(n) = O(n^k)$.
   (b) $f(n) = \Omega(n^k)$.
3. Order the following functions by growth rate: N, $\sqrt{N}$, $N^{1.5}$, N^2, $N \log N$, $N \log\log N$, $N \log^2 N$, $N\log (N^2)$, $2/N$, 2^N, $2^{N/2}$, 37, $N^2 \log N$, N^3. Indicate which functions grow at the same rate.
4. Suppose $T_1(N) = O(f(N))$ and $T_2(N) = O(f(N))$. Which of the following are true ?
   a. $T_1(N) + T_2(N) = O(f(N))$
   b. $T_1(N) - T_2(N) = o(f(N))$
   c. $T_1(N) / T_2(N)= O(1)$
   d. $T_1(N) = O(T_2(N))$.
5. Which functions grows faster: $N \log N$ or $N^{1+\epsilon/\sqrt{\log N}}$, $\epsilon > 0$?
6. Prove that for any constant, k, $\log^k N = o(N)$.
7. Find two functions f(N) and g(N) such that neither $f(N) = O(g(N))$ nor $g(N) = O(f(N))$.
8. For each function f(n) and time t in the following table, determine the largest size n of a problem that can be solved in time t assuming that the algorithm to solve the problem takes f(n) microseconds.

	1 second	1 hour	1 Month	1 century
log n				
$\sqrt{n}$				
N				
n log n				
n^2				
n^3				
2^n				
n !				

9. Show that $\log^3 n$ is $o(n^{1/3})$.
10. Show that $\Sigma^n_{i=1} i^2$ is $O(n^3)$.
11. Consider the following functions of n:
    $f_1(n) = n^2$
    $f_2(n) = n^2 +1000n$
    $f_3(n) = n$, if n is odd
    n^3, if n is even.

    $f_4(n) = n$, if $n <= 100$
    n^3, if $n > 100$
    Indicate for each distinct pair i and j whether $f_i(n)$ is $O(f_j(n))$ and whether $f_j(n)$ is $\Omega(f_i(n))$.
12. Consider the following functions of n:
    $g_1(n) = n^2$ for even $n >=0$
    n^3 for odd $n >=1$

    $g_2(n) = n$, $0 <= n <=100$
    n^3, for $n > 100$
    $g_3(n) = n^{2.5}$

    Indicate for each distinct pair i and j whether $g_i(n)$ is $O(g_j(n))$ and whether $g_j(n)$ is $\Omega(g_i(n))$.

13. Give, using "bog oh" notation, the worst case running times of the following procedures as a function of n.

a.
```
procedure matmpy ( n : integer);
var
        i, j, k : integer;
begin
        for I = 1 to n
                for j = 1 to n do begin
                        C[I,j] = 0
                        for k = 1 to n do
                        C[I, j] = C[I,j] +A[I, k]*B[k,j]
                end
end
```

b.
```
procedure mystery (n: integer)
var
        i,j,k : integer;

begin
        for i = 1 to n-1 do
        for j= i+1 to n do
                for k = 1 to j do
                        {some statement requiring O(1) time}
end
```

c.
```
procedure veryodd ( n : integer);
var
        i, j, x, y : integer;
begin
        for i = 1 to n
                if odd(i) then begin
                for j = i to n do begin
                x = x+1;

                for j= 1 to ido
                y = y +1;
        end
end
```

d.
```
function recursive(n:integer) : integer;
begin
        if n <= 1 then
                return(1)
        else
                return(recursive(n-1) + recursive(n-1))
end
```

14. Show that the following statements are true.
   a. 17 is O(1).
   b. n(n-1) is $O(n^2)$
   c. $\sum^{n}_{i=1} i^k$ is $O(n^{k+1})$ and $\Omega(n^{k+1})$ for integer k.
   d. If P(x) is any k^{th} degree polynomial with a positive leading coefficient, then P(n) is $O(n^k)$ and $\Omega(n^k)$.

15. Order the following functions by growth rate: n, sqrt(n), log n, log log n, $\log^2 n$, n/log n, $\sqrt{n}\log^2 n$, $(1/3)^n$, $(3/2)^n$, 17.