Assignment - 4
Analysis of Algorithms

1) Arrange the following expressions by growth rate from slowest to fastest.

$4n^2$, $\log_3 n$, $n!$, 3^n, $20n$, 2, $\log_2 n$, $n^{2/3}$

Use Stirling's approximation in for help in classifying n!

Stirling's approximation states that $n! \approx \sqrt{(2 \pi n)}\ (n/e)^n$

2) Estimate the number of inputs that could be processed in the following cases:

(a) Suppose that a particular algorithm has time complexity $T(n) = 3 \times 2^n$, and that executing an implementation of it on a particular machine takes t seconds for n inputs. Now suppose that we are presented with a machine that is 64 times as fast. How many inputs could we process on the new machine in t seconds?

(b) Suppose that another algorithm has time complexity $T(n) = n^2$, and that executing an implementation of it on a particular machine takes t seconds for n inputs. Now suppose that we are presented with a machine that is 64 times as fast. How many inputs could we process on the new machine in t seconds?

(c) A third algorithm has time complexity $T(n) = 8n$. Executing an implementation of the algorithm on a particular machine takes t seconds for n inputs. Given a new machine that is 64 times as fast, how many inputs could we process in t seconds?

3) A hardware vendor claims that their latest computer will run 100 times faster than that of their competitor. If the competitor's computer can execute a program on input of size n in one hour, what size input can vendor's computer execute in one hour for each algorithm with the following growth rate equations?

$n \qquad n^2 \qquad n^3 \qquad 2^n$

4) For each of the following pairs of functions, either f(n) is in O(g(n)), f(n) is in Ω(g(n)), or f(n) = Θ(g(n)). For each pair, determine which relationship is correct. Justify your answer.

(a) $f(n) = \log n^2$; $g(n) = \log n + 5$.
(b) $f(n) = \sqrt{n}$; $g(n) = \log n^2$.
(c) $f(n) = \log 2^n$; $g(n) = \log n$.
(d) $f(n) = n$; $g(n) = \log n$.
(e) $f(n) = n \log n + n$; $g(n) = \log n$.
(f) $f(n) = \log n^2$; $g(n) = (\log n)^2$.

(g) $f(n) = 10$; $g(n) = \log 10$.
(h) $f(n) = 2^n$; $g(n) = 10n^2$.
(i) $f(n) = 2^n$; $g(n) = n \log n$.
(j) $f(n) = 2^n$; $g(n) = 3^n$.
(k) $f(n) = 2^n$; $g(n) = n^n$.

6) Determine Θ for the following code fragments in the average case. Assume that all variables are of type **int**.

(a)
```
a = b + c;
d = a + e;
```

(b)
```
sum = 0;
for (i=0; i<3; i++)
    for (j=0; j<n; j++)
        sum++;
```

(c)
```
sum=0;
    for (i=0; i<n*n; i++)
        sum++;
```

(d)
```
for (i=0; i < n-1; i++)
    for (j=i+1; j < n; j++) {
        tmp = AA[i][j];
        AA[i][j] = AA[j][i];
        AA[j][i] = tmp;
    }
```

(e)
```
sum = 0;
for (i=1; i<=n; i++)
    for (j=1; j<=n; j*=2)
        sum++;
```

(f)
```
sum = 0;
for (i=1; i<=n; i*=2)
    for (j=1; j<=n; j++)
        sum++;
```

(g) Assume that array A contains n values, **random** takes constant time, and **sort** takes n log n steps.
```
for (i=0; i<n; i++) {
    for (j=0; j<n; j++)
        A[i] = util.random(n);
        sort(A);
    }
```

(h) Assume array A contains a random permutation of the values from 0 to n - 1.

```
sum = 0;
for (i=0; i<n; i++)
    for (j=0; A[j]!=i; j++)
        sum++;
```

(i)

```
sum = 0;
if (EVEN(n))
    for (i=0; i<n; i++)
        sum++;
    else
        sum = sum + n;
```