Assignment 2

1. **Show that function log2(9N) is O(log2N), You will need to use the definition**
   of O(f(n)) to do this. In these words, find values for c and n0.

Solution: let $f(n) = (9N)$ and $g(n) =$ O(log2N)

Prove f(n) is O(g(n)) = O(log2N)

$f(n) = \ (9N)$

$= \ (9) \ + (N)$

$\leq C_0 N$

Hence,

$\exists c_0 = 2$, $n_0 = 9$ $such\ that\ when\ n\ n_0,\ f(n) \leq$ $c_0 N$ $always\ holds\ true.$

2. **Order the following functions by the growth rate:**

$N^2, NlogN, \frac{2}{N}, 2^N, 92, N^2 logN, N!, N^{1.5}, N^3, logN, Nlog\left(N^2\right), 4^{logN}, N. Indicate$

$which\ function\ grow\ at\ the\ same\ rate.\ A\ function\ f(N) grows\ at\ the\ same\ rate\ as$

$function\ \ g(N) if\ f(N) = \ \theta(g(N))$

Solution:

The growth order is $\frac{2}{N}, 92, logN, N, NlogN, NlogN^2, N^{1.5}, N^2, 4^{logN}, N^2 logN, N^3, 2^N, N!$

The functions whose growth rate is same :

The functions N and N grow at same rate

N*log(N^2) = 2N*log N

When taking order of N:

2N * log(N) = O(N log N)

Proof:

$Let's\ assume\ that\ 4^{logN} =$ N^{logN} $then\ log4\ will\ have\ different\ values\ when\ the\ base\ value$

$let's\ take\ different\ bases\ to\ see\ the\ growth\ rate.\ Now,\ we\ take\ the\ base\ values\ like\ base2,$

$base\ 4.$

$case\ 1: Base\ =\ 2\ then\ 4\ \ =\ 2$

$then\ the\ growth\ rate\ is\ as\ follows:$

$\frac{2}{N}, 92, logN, N, NlogN, NlogN^2, N^{1.5}, N^2, 4^{logN}, N^2 logN, N^3, 2^N, N!$

$case\ 2: Base\ =\ 4\ then\ 4\ = 1$

$then\ the\ growth\ rate\ is\ as\ follows:$

$\frac{2}{N}, 92, logN, N, NlogN, NlogN^2, N^{1.5}, N^2, 4^{logN}, N^2 logN, N^3, 2^N, N!$

$Hence,\ as\ all\ the\ 2\ cases\ have\ same\ growth\ So,\ the\ growth\ rate\ is\ correct.$

3. $Suppose\ T_1(N) = O(f(N))\ and\ T_2(N) O(f(N)).\ which\ of\ the\ following\ are\ true.$

Solution :

$a.\ T_1(N) + T_2(N) = O(f(N)): True$

$b. T_1(N) - T_2(N) = o(f(N)): False$

$Example: Let's assume\ that\ T_1(N) = 2N, T_2(N) = N,\ f(N) = N then$

$T_1(N) - T_2(N)$ = 2N – N

=N => $\theta(N)$

=$\theta f((N))$

Therefore, $T_1(N) - T_2(N)$ ≠o(f(n)).

$c. \frac{T_1(N)}{T_2(N)} = O(1): False$

$Example: let's assume\ that\ T_1(N) = N^4 + N^3,\ T_2(N) = N^2\ So,$

$$\frac{T_1}{T_2} = \frac{N^4+N^2}{N^2} = N^2 + 1\ which\ has\ a\ O(N^2) \neq O(1)$$

d. $T_1(N) = O(T_2(N)): False$

$Example:\ T_1(N)$ = N^2 , $T_2(N)$ = N and f(N) = N^2

Clearly, $N^2 \neq O(N)$

4. **Give the Big Oh for each of the following code excerpts. For Part a-c, verify your Big Oh doing a precise algorithm analysis, using summations and reducing to close forms as demonstrated in class. You may want to refer to section 1.2.3 in the book for series formulas. For full credit show your work of how you used summation to reduce to closed form. For part d give a brief explanation as to how you came up with your Big Oh.**

a) $sum = 0;$

$for\ (i = 1; i \leq n; i\ ++)\{$

$for(j = 1; j \leq n; j\ ++)\{$

$sum\ ++;$

Sum ++;

}

}

Solution: $sum = 0;$ //O(1)

$for\ (i = 1; i \leq n; i\ ++)\{$ //loop executes O(n) times

$for(j = 1; j \leq n; j\ ++)\{$ //executes O(n) times

$sum\ ++;$

Sum ++; (does 2x increment)

}

}

So, $O(1) + O(n * n) = O(N^2)$

Using summation:

The first for loop is like $\sum_{i=1}^{n}$

The second for loop gives $\sum_{j=1}^{n} 2$

So, $\sum_{i=1}^{n} (\sum_{j=1}^{n} 2) = \sum_{i=1}^{n} 2N$

$= 2N \sum_{i=1}^{n} 1$

$= 2N * N$

$= 2N^2$

$= \theta(N^2)$

b) Sum = 0

For(i=1 ; I <= n; i++){ //O(n)

For(j=1 ; j < = 3 * i ; j++) { //O(n)

Sum ++;

```
    }
    For(k =1;k <= 100000; k++) {
    Sum++;
    }
}
```

Solution:

So, O(1) + O(N * N) = $O(N^2)$

Using Summation:

The first for loop represents $\sum_{i=1}^{n}$

The second for loop represents $\sum_{j=1}^{3i} 1$

The third loop represents $\sum_{k=1}^{100000} 1$

So, $\sum_{i=1}^{n} (\sum_{j=1}^{3i} 1 + \sum_{k=1}^{100000} 1)$

$= \sum_{i=1}^{n} (3i + 100000)$

$= \sum_{i=1}^{n} 3i + \sum_{i=1}^{n} 100,000$

$=3\, n(n + 1)/2 + 100,000n$

$= O\left(N^2\right)$

c)

```
Sum = 0
For(i= 1; I <= n; i++) {  //O(n)
    For(j=1;j<=I * i; j++){  //O(n^2)
        Sum ++;
  }
}
```

Solution:

So, $O(N * N^2) = O(N^3)$

Using summation:

The first for loop gives $\sum_{i=1}^{n}$

The second for loop represents $\sum_{j=1}^{i\wedge 2} 1$

$$= \sum_{i=1}^{n} (\sum_{j=1}^{i^2} 1)$$

$$= \sum_{i=1}^{n} i^2 = 1 + 4 + 9 ..... + n^2$$

$$= n(n + 1)(2n + 1)/6$$

$$= \theta\left(N^3\right)$$

d)
```
Sum = 0;
For(i=1;i<=n; i++){
     For(j=1 ; j <=i * i; j++){
              If(j%i == o){
                 For(k=1;k<=j; k++){
                        Sum ++;
                 }
              }
     }
}
```

Solution:
```
Sum = 0;   //O(1)
For(i=1;i<=n; i++){                    //O(n)
     For(j=1 ; j <=i * i; j++){             //O(n^2)
              If(j%i == o){
                 For(k=1;k<=j; k++){     //O(n)
                        Sum ++;
                 }
              }
     }
}
```

So, $O(1) + O(N*N^2*N)$

$= O(N^4)$

Using summation:

The first for loop gives $\sum_{i=1}^{n}$

The second for loop represents $\sum_{j=1}^{i^2} 1$

The third for loop represents $\sum_{k=1}^{j} 1$

So, $\sum_{i=1}^{n} (\sum_{j=1}^{i^2} 1 + \sum_{k=1}^{j} 1)$

$$\sum_{i=1}^{n} (N^2 + 1)$$

$$= \mathrm{n}(N^2 + 1)$$

$$= N^3 + \mathrm{N}$$

$$= N^4$$

$$= \mathrm{O}(N^4)$$

5. Electronic submission.