Homework 1

(75 points)

Submit: eLearning

Late Policy: -10 points per hour late

Instructions: Include Name and Net ID. This is an individual assignment. Answers should be your own work. ANDY NGUYEN ADN200004

1. Order the following functions by growth rate **(12 points)**

   2^{50}
   N
   $\sqrt{N}$
   $N^{1.5}$
   N^2
   NlogN
   NloglogN
   $Nlog^2N$
   $Nlog(N^2)$
   2/N
   2^N
   37
   N^2logN
   N^3

   2
   N
   37, 2^{50} ← same growth rate
   $\sqrt{N}$
   N
   NloglogN
   NlogN
   Nlog(N²), NlogN²
   $N log^2 N$
   $N^{1.5}$
   N^2
   $N^2 logN$
   N^3
   $2^{N/2}$
   2^N

   Indicate which of the functions grow at the same rate.

2. Give the Big-O notation for the following expressions: **(10pts, 2pt each)**

   a. $2n^4 + 3n^3 - 5$ $O(n^4)$

   b. $4^n - n^2 + 19$ $O(4^n)$

   c. $\sum n$ $O(n)$

   d. $3n * log(n) + 8$ $O(n \log n)$ $n(n+1)$ n^2

   e. $[n(n+1)/2 + 2n]^2$ $O(n^2)$

3. For each of the following code fragments give running time analysis (Big Oh). Explain your answer **(25pt, 5pts each)**

a.
```
sum =0;
for ( i=0; i < n; i++)
  sum++;
```
O(n), worst case the loop runs n times

b.
```
sum =0;
for( i = 0; i < n; i++)
  for(j = 0; j < i ; j++)
       sum++;
```
$O(n^2)$, worst case scenario i has to run n times and since j is nested, j also has to run n times

c.
```
sum =0;
for( i = 0; i < n; i++)
  for( j = 0; j < i *i ;  j++)
       for( k = 0; k<j; k++)
            sum++;
```
$O(n^5)$, worst case i run n times. Inner loop runs n^2 times. Inner inner loop runs n^2 times. $n^2 \cdot n^2 \cdot n = n^5$

d.
```
if(value < n)
  for( i = 0; i < n; i++)
  {
       System.out.println(i);
  }
else
  System.out.println(value);
```
O(n), worst case i will run n times. The else branch is constant time

e.
```
sum2 = 0;
sum5 = 0;

for(i=1; i<=n/2; i++)
{
  sum2 = sum + 2;
}
for(j=1; j<=n*n; j++)
{
  sum5 = sum + 5;
}
```
$O(n^2)$, cause we have two non-nested loops

$n/2$ n^2

$n^2 + \frac{n}{2}$, ignore lower order we get $n^2 + \frac{n}{2}$

4. What is the time complexity of the below function? **(10 points)**

```
void fun(int n, int arr[])
{
    int i = 0, j = 0;
    for(; i < n; ++i)
        while(j < n && arr[i] < arr[j])
```
O(n), cause j will run n times and since it is not being set back to 0 each iteration of outer loop

```
            j++;
    }
```

5. **What is the Big-O running time for this code? Explain your answer. (10 points)**

```
int i = numItems;
while (i > 0)
{
     i = i / 2;  // integer division will eventually reach
zero
}
```

O(log2(n)), because items are being split in half each iteration

6. An algorithm takes 0.5 ms for input size 100. How long will it take for input size 500 if the running time is the following (assume lower order terms are negligible) **(8 Points)**
   a. linear
   b. O(N log N)
   c. quadratic
   d. cubic

0.5ms per 100 $0.5 \cdot 5 = 2.5ms$

1 input size
0.005ms

$\frac{100}{0.5} = \frac{1}{0.005}$

$\frac{0.5}{100} = \frac{}{500}$

$\frac{0.5}{100\log(100)} = \frac{}{}$

b) $\frac{0.5}{100\log(100)} = \frac{?}{500\log(500)}$

$?= 3.3737 ms$

c) $\frac{0.5}{100^2} = \frac{?}{500^2}$

$?= 12.5ms$

d) $\frac{0.5}{100^3} = \frac{?}{500^3}$

$?= 62.5ms$