MOdule 7 Activity 1

# Analysis of algorithms

### Concepts Covered

* Big O notation
* Order of the algorithm – Asymptotic complexity

### Activity Instructions: Answer the following questions in groups.

1. **What is the order (asymptotic complexity) of the following growth functions?** 
   1. 10n2 + 100n + 1000
   2. 10n3 – 7
   3. 2n + 100n3

**Answer**: b, a, c

1. **Below is the code to find the largest element in an unsorted array of integers. What is the time complexity Big (0) of this algorithm? Assume the length of the array is n.**

int max;

if (int Array.length > 0) //

{

max = int Array[0];

for (int num = 1; num < int Array.length; num++) //

if (int Array[num] > max) //

max = intArray[num];

System.out.println (max);

}

else

{

System.out.println ("The array is empty.");

}

**Answer**: O(n)

1. **Determine the growth function and order of the following code fragment:**

for (int count = 0; count < n; count ++)

{

for (int count2 = 1; count2 < n; count2 = count2 \* 2)

{

//function call with O(1) complexity;

}

}

Answer: O(n^2)

**Challenge:**

1. **Determine the best case and worst case (*Big-O ) time complexity for the following operations of an array of length n*:** 
   1. **Access: Best = O(1), Worst = O(n)**
   2. **Insertion: Best = O(n), Worst = O(log(n))**
   3. **Deletion: Best = O(1), Worst = O(n)**
   4. **Search: Best = O(1), Worst = O(n)**

### Deliverable

* Submit your worksheet on Canvas