ANYANG TECHNOLOGICAL UNIVERSITY
School of Electrical & Electronic Engineering

IE2108 Data Structures and Algorithms

Tutorial No. 3 (Sem 1, AY2022-2023)

1. Determine the order of growth of the following sums. Use the $O(g(n))$ notation with the *simplest* function $g(n)$ possible.
   (i) $\sum_{i=0}^{n-1}(i^2+1)^2$
   (ii) $\sum_{i=2}^{n-1}\lg i^2$
   (iii) $\sum_{i=0}^{n-1}\sum_{j=0}^{i-1}(i+j)$

2. The algorithm for finding the maximum element of an array is shown as follows:

```
Input: array A of n integers
Output: maximum element of A

Algorithm arrayMax(A, n)
currentMax = A[0]
for i = 1 to n−1
  if A[i] > currentMax
    currentMax = A[i]
return currentMax
```

   Determine the number of times that the statement “`currentMax = A[i]`” will be executed in the best case and in the worst case.

3. For each of the following algorithms, give an asymptotic notation for the number of times that the statement $x = x + 1$ is executed.

(i)
```
for i = 1 to n
 for j = 1 to i
    for k = 1 to j
    x = x + 1
```

(ii)
```
j = n
while (j ≥ 1) {
  for i = 1 to j
   x = x + 1
  j = j/3
}
```

4. Find the first 4 terms of the recurrence relation $a_k = 2a_{k-1} + k$, where $a_1 = 1$.

5. Solve the recurrence relation to compute the value for a_n: $a_n = a_{n-1} + 3$, where $a_1 = 2$.

6. Determine the complexity of the following recursive function. (You may assume that $n = 2^k$).

$$T(n) = 2T\left(\frac{n}{2}\right) + cn \qquad \text{if } n > 1$$

$$T(n) = 1 \qquad \text{if } n = 1.$$

7. Consider the following recursive algorithm,

```
Input: positive integer n
Output: Q(n)

Algorithm Q(n)
  if n = 1
     return 1
  else
     return Q(n-1) + 2*n - 1
```

Set up a recurrence relation for the *number of multiplications* made by the algorithm and solve it.