

Tutorial 3

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 in the following

```

Algorithm arrayMax(A, n)
Input array A of n integers
Output maximum element of A
currentMax = A[0]
for i = 1 to n − 1 do
    if A[i] > currentMax then
        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 algorithm, give an asymptotic notation for the number of times which the statement $x=x+1$ is executed.

(i)
$$\begin{array}{l} \text{for } i = 1 \text{ to } n \\ \quad \text{for } j = 1 \text{ to } i \\ \quad \quad \text{for } k = 1 \text{ to } j \\ \quad \quad \quad x=x+1 \end{array}$$

(ii)
$$\begin{array}{l} \text{for } i = 1 \text{ to } 2n \\ \quad \text{for } j = 1 \text{ to } n \\ \quad \quad x=x+1 \end{array}$$

(iii)
$$\begin{array}{l} j = n \\ \text{while } (j \geq 1) \{ \\ \quad \text{for } i = 1 \text{ to } j \\ \quad \quad x=x+1 \\ \quad j = j/3 \\ \} \end{array}$$

4. Solve the following recurrence relation to compute the value for a_n : $a_n = a_{n-1} + 3$,
 $a_1 = 2$
5. Determine the complexity of the following recursive function. (You may assume that $n=2^k$).
$$T(n) = 2 T(n/2) + cn \quad \text{if } n > 1$$
$$T(n) = 1 \quad \text{if } n = 1.$$
6. Consider the following recursive algorithm.
Algorithm $Q(n)$
Input: positive integer 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.
-