**7SENG010W Data Structures & Algorithms**

**Week 1 Tutorial Exercises**

These exercises cover: Abstract Data Types (ADTs), Big-O Complexity, Timing an Algorithm

**Exercise 2.**

Complete the table for the quadratic growth rate running time equation T(N) = 2N2 + 3N + 4. This gives an idea why when calculating the Big-O for a T(N) that only the value of the *dominant term*, i.e. N**2**, determines its order of complexity.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Values of N** | **T(N) = 2N2 + 3N + 4** | | | |
| **2N2** | **3N** | **4** | **T(N)** |
| **1000** |  |  |  |  |
| **2000** |  |  |  |  |
| **4000** |  |  |  |  |
| **8,000** |  |  |  |  |
| **16,000** |  |  |  |  |

**Exercise 3.**

Complete the table for the following common growth rates for Big-O for the values of N. To help you calculate these values see the list of online tools given in the module's *Software Requirements* document. Note that it may not be possible to calculate the value of some of these expressions for large N, in those cases experiment to find large values of N that they can be evaluated for.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Big-O** | **Values of N** | | | | |
| **20** | **50** | **100** | **1000** | **100,000** |
| O(1) |  |  |  |  |  |
| O(N) |  |  |  |  |  |
| O(N**2**) |  |  |  |  |  |
| O(N**3**) |  |  |  |  |  |
| O( log2(N) ) |  |  |  |  |  |
| O( N log2(N) ) |  |  |  |  |  |
| O( 2**N** ) |  |  |  |  |  |
| O( N! ) |  |  |  |  |  |

Note that when we deal with order of complexity expressions that involve log**2**(N) the result is very rarely a whole number, so the standard practice is to take the smallest whole number that is greater than the fractional value, this is called the “ceiling” in maths e.g. ceiling( 4.3219 ) = 5.

For comparison, the estimated age of the Universe is approximately 13.5 billion years, & in seconds that is: (13.5 × 10**9**) × ( 365×24×3600) = 4.25736×10**¹⁷**.