**Time Complexity Lab 2 Array task with Expected Worst case time complexity O(n)**

**Document Id 9**

**Questions**

1. An algorithm is made up of two independent time complexities f (n) and g (n). Then the complexities of the algorithm is in the order
2. f(n) + g(n)
3. f(n) x g(n)
4. Max( f(n), g(n))
5. Min( f(n), g(n))

2) Big O notation is defined for

1. Time and Space complexity
2. Optimality
3. Array Operations
4. Searching

3) Hash tables are a good choice when you need

1. Fast lookup
2. Small data size
3. You want a key value pair
4. Fast Insertion

**Practical Array task with Expected Worst case time complexity O(n)**

Using the result that the [sum of consecutive integers from 1 to n is equal to n(n+1)/2](http://en.wikipedia.org/wiki/1_%2B_2_%2B_3_%2B_4_%2B_%E2%8B%AF).

Create an algorithm that finds the missing element in a arithmetic sequence with a constant difference of 1.

Example 1, 2, 3, 5 Ans missing element is 4

.