ITEC415: Analysis of Algorithms
Recursive, Iteative Formulations

- ***By using recursive approach perform the following <u>first 3 problems</u> by using functions in Java***
- ***Anlyse their working mechanisms and check their correctness***
- ***Comment on their T(n)***

1. Factorial calculation can be defined as follows

$$n! = \begin{cases} 1 & \text{If } n=0 \\ n(n-1)! & \text{If } n>0 \end{cases}$$

Ex:

3!= 3*2!
=3*2*1!
=3*2*1*0!
=3*2*1*1
=6

2. The derivation of Fibonacci numbers

1, 1, 2, 3, 5, 8, 13, 21, …

$$FIB(N) = \begin{cases} FIB(N-1)+FIB(N-2) & \text{If } N>2 \\ 1 & \text{If } N=2 \\ 1 & \text{If } N=1 \end{cases}$$

3. The finding of greatest common divisor by using of Euclid's algorithm

$$GCD(m,n) = \begin{cases} GCD(n,m) & \text{If } n > m \\ m & \text{If } n = 0 \\ GCD(n, \text{mod}(m,n)) & \text{If } n > 0 \end{cases}$$

4. Trace the following sorting algorithm witn a small number sets [10 7 23 5 20]

```
SELECTION-SORT(A)
n = A.length
for j=1 to n-1
    smallest= j
    for i = j+1 to n
        if A[i] <
            A[smallest]
            smallest= i
    exchange A[j]    with A[smallest]
```

- Analyse and find out, what is the running time of the algorithm for all cases.

5. There are two ways of binary search algorithms given below. Trace, analyse and find out what is T(n) on both?

```
ITERATIVE-BINARY-SEARCH(A, v, low, high)
while low ≤ high
    mid = ⌊(low + high)/2⌋
    if v == A[mid]
        return mid
    elseif v > A[mid]
        low = mid + 1
    else high = mid − 1
return NIL

RECURSIVE-BINARY-SEARCH(A, v, low, high)
if low > high
    return NIL
mid = ⌊(low + high)/2⌋
if v == A[mid]
    return mid
elseif v > A[mid]
    return RECURSIVE-BINARY-SEARCH(A, v, mid + 1, high)
else return RECURSIVE-BINARY-SEARCH(A, v, low, mid − 1)
```