**AC21005 – Data Structures and Algorithms**

**Assignment 1**

**Problems**:

1. Find on the web the Euclidean algorithm of computing the greatest common divisor.

Program it in a language of our choice.

2. Define **for** - loop using **while**-loops (in pseudo-code and in programming language of your choice).

3. The factorial function *n!* has value *1* when *n≤1* and value *n\*(n-1)!* when *n>1*. Write both a recursive and iterative algorithm to compute *n!* .

Determine their space complexity.

4. Prove by induction correctness of the algorithm RSum.

5. Introduce time-step “counter” to the Recursive Sum algorithm (from Lecture 3 and Lecture 4 (the table on p.6)).

6. Analyse Time complexity of SelectionSort algorithm from Lecture 2 :

a) Using Method 1 or 2 for counting time steps.

b) Using Big-O notation.

7. (\*) Prove by induction:

**Solutions:**1. Program coded in Java. File included.

2. Pseudocode:  
 Ask user for the amount of times they wish to loop for  
 Store as an integer variable  
 While users number is greater than 0  
 Do something  
 Decrement x value

Next loop iteration

Program coded in Java. File included.

3.

|  |  |
| --- | --- |
| **Recursion** | **Iteration** |
| *RecursiveFactorial(number)*  {  if*(number≤0)* then return *0.0*  else return number\**RecursiveFactorial(number-1)*  } | *IterativeFactorial(number)*  {  Temp = 1  while number > 1  temp=temp\*number  number--  return temp  } |
| **Space Complexity**  N+1 | **Space Complexity**  2 |