Mila University Center
Institute of Science and Technology

2nd year LMD computer science
ASD3 Module
Year: 2022-2024

TD Series N 01
Complexity

Exercise 1: Give, as a function of N, the number of assignments, additions, multiplications and comparisons carried out by the following function:

```
Function Power (A, N : integer): integer
        i, R: integer.
Begin
   i ←0 ; R ←1;
   While ( i < N )do
      R← R * A;
      i ←i+1 ;
   End while
   Return R;
END
```

Exercise 2: Give, as a function of N, the number of assignments (R ← R + 1) performed by the algorithm below? What does this algorithm calculate? Give, if possible, two algorithms faster than this algorithm.

```
Algorithm Example
        I, J, N, R: integer
Begin
   R ←0 ;
   For I ← 1 to N do
      For J ← 1 to I do
        R← R + 1 ;
      End For
   End For
   Write (R)
END
```

Exercise 3: What does the following program do? Evaluate its complexity.

```
int main ( )
{       float price, quantity , amount;
        cin >> price >> quantity ;
        amount = price * quantity;
        if ( quantity <= 350)
              cost << amount;
        else
             if ( quantity > 350 && quantity <=600 )
             {
                        amount=amount-amount*0.02 ;
                        cost << amount;
             }
            else
                       cost << amount- amount*0.03;}
```

Exercise 4: Write the function that returns the maximum value of a matrix of n*m integers. How much time it will take to find the minimum value of a matrix of $10^3 * 10^3$ elements. We consider that the speed of the used machine is 10^6 operations per second.

Exercise 5: give the complexity classes of the following functions

1. $F(n) = 3n + 2\log(n) + 10$
2. $T(n) = 3n^2 + 5$
3. $C(n) = n \log(n) + 8n + 3$
4. $P(n) = 2^n + 10n^3 + 8$
5. $Q(n) = 15$

Exercise 6: evaluate the worst-case complexity for the function search below, which searches for an element in a sorted array. Compare this complexity with the complexity of a function based on a sequential search.

Function dichotomous_search (T:array of integers, x, N: integer): boolean
inf , sup, m: integer; b: boolean
Begin
inf ←1; sup ←N; b ←false;
While (inf <= sup and b=false) do
m ←(inf + sup) / 2;
if (x = T[m]) then
B ←true ;
else
if (x < T[m]) then
sup ←m - 1;
else
inf ←m + 1;
End If
End if
End While
Return b;
END

Exercise 7: Evaluate the complexity of the following recursive function.

```
int power ( int x, int n)
{
    if (n==0)
        return 1;
    if (n==1)
        return x;
    else
        return x * power(x, n-1);
}
```

Exercise 8: Write an algorithm that allows you to enter an array of integer and then display the sum of its elements. Evaluate its complexity.