Data Structures and Algorithms

WEEK1

2ND YEAR - ENSIA

OBJECTIVES

- Proof by Induction vs. Counter-example vs. Contradiction
- Recursion vs. Iteration + Introducing notion of Computational time/space

EXERCISES

Exercise 1:

Prove the following Formulas:

- $log(A^B) = B\ log\ A$
- $log\ X\ <\ X\ for\ all\ \ X\ >0$

Exercise 2:

Using **Induction**, prove the following formulas:

- $\sum_{i=1}^{N} \quad (2i-1) = N^2$
- $\sum_{i=1}^{N} \quad i^3 = (\sum_{i=1}^{N} \quad i)^2$
  You can make use the formula : $\sum_{i=1}^{N} \quad i^3 = N^2(N+1)^2\ /4$

Exercise 3:

Let F_i be the Fibonacci numbers defined as $F_{k+1}=F_k+F_{k-1}$ such that $F_0=1$ and $F_1=1$

Prove the following:

- $\sum_{i=1}^{N-2} \quad F_i = F_N - 2$
- $F_N < \phi^N$ such that $\phi = (1+\sqrt{5})/2$ (ϕ is known as the Golden Ratio)

Exercise 4:

Proof by Contradiction that For all integers n, if n^3+5 is odd then n is even

Exercise 5:

Write a recursive **algorithm** that returns the number of 1s in the binary representation of *N*. Use the fact that this is equal to the number of 1 in the representation of *N/2* when N is even. If N is odd, there is an additional 1.

- (Home) : What is the number of zeros for a given binary number ?

Exercise 6 :

Design efficient algorithms that take an array of positive numbers a , and determine:

1. The maximum value of a[j]+a[i] , with j ≥ i .
2. The maximum value of a[j]-a[i] , with j ≥ i .
3. The maximum value of a[j]*a[i] , with j ≥ i .
4. The maximum value of a[j]/a[i] , with j ≥ i .

Exercise 7:

Evaluate the following sums:

- $\sum_{i=0}^{\infty} \frac{1}{4^i}$
- $\sum_{i=0}^{\infty} \frac{i}{4^i}$
- $\sum_{i=0}^{\infty} \frac{i^2}{4^i}$
- $\sum_{i=0}^{\infty} \frac{i^N}{4^i}$

Exercise 8:

Write a Pseudocode for computing the Fibonacci number using both paradigms :

- Iterative
- Recursive

(Home) which one takes more time to compute for a given number N (Example N=100 000)

Exercise 9 (Home):

Given a decimal number (For instance : 34892), write a recursive function to flip the number from right to left (→ 29843) (Please, don't use *String* functions)

Exercise 10 (Home):

For a given string (Example , "ABC"), write a recursive function to print to the console all possible permutations of the given string (→ ABC ACB BAC BCA CBA CAB)

Exercise 11 (Home):

For a vending machine to return the change to the customer, it must return a minimal number of coins. Write a recursive function to determine the number of each type of coin to give back to the customer as part of the change.
(For example : 290 → 2x100DA + 1x50DA + 2x20DA + 1x10DA)