**INTRODUCTION TO ALGORITHMS**

3-2-0-4

Instructor: Naresh Jotwani, [naresh\_jotwani@daiict.ac.in](mailto:naresh_jotwani@daiict.ac.in)

Reference: **INTRODUCTION TO ALGORITHS**, by Cormen, Leiserson, Rivest & Stein, PHI. [//mitpress.mit.edu/algorithms](file:///mitpress.mit.edu/algorithms) in short, [CLRS].

Stress will be on the practical aspects of algorithms.

Attendance is NOT compulsory -- but regularity is expected.

Classes: Wednesdays, 12:00 noon. Fridays: 10:00 am & 12:00 noon.

\*\*\*

Based on [CLRS]:

... an algorithm is a *well-defined computational procedure* that takes as ***input*** a *set of values* and produces as ***output*** another *specified* *set of values* ...

... a sequence of computational steps that transforms *as specified* the ***input*** into ***output***.

Clearly, we need a notation to define – that is, describe – an algorithm. We shall use the notation of ***pseudocode*** for this purpose.

\*\*\*

Suppose we need to produce in output the ***sorted*** set of values in input.

Question: When is it possible to sort a set of values?

Hint: Can we compare apples and oranges? An ordering relation must exist.

Let us consider a simple sorting algorithm.

**INSERTION-SORT(A)**

**for j = 2 to length[A]**

**key = A[j];**

**// Insert A[j] into sorted sequence A[1..j-1]**

**i = j-1;**

**while i > 0 and A[i] > key**

**A[i+1] = A[i];**

**i = i-1;**

**A[i+1] = key;**

NOTE:

1. The notation used is different from that used in [CLRS]. The above notation of pseudocode does not include **do**, curly brackets ... *et cetera.*

2. Red font 🡪 elements are being compared.

3. Importance of indentation. Note the differences.

4. This sort algorithm works in-place.

5. Learn the importance of testing for end values of loop variable.

**Loop invariant: Every time we enter the for loop, the sequence**

**A[1..j-1] is sorted.**

VERY IMPORTANT: Analysing algorithms ‘on paper’ – that is, even when one is not programming the algorithm.

**QUESTION 1**: How much time will this algorithm take for sorting 10, 100, 1000 ... 106 elements?

**QUESTION 2**: Do we have enough information to answer the above question?

Think about such questions related to algorithms.