# Fundamentals of Algorithms
# CS502-Fall 2011

# ASSIGNMENT #2

## *Deadline*

Your assignment must be uploaded/submitted at or before **21st Nov. 2011**

## *Uploading instructions*

Please view the **assignment submission process** document provided to you by the Virtual University to upload the assignment.

## *Rules for Marking*

It should be clear that your assignment will not get any credit if:

- The assignment is submitted after due date.
- The submitted assignment does not open or run.
- **The assignment is copied.**

## *Objectives*

This assignment will help you to understand the concept of recurrence relations and way to solve then and writing asymptotic notation after analyzing and solving recurrences .The other main focus is to learn dynamic programming applications and edit distance problem solution using dynamic programming technique which will be ultimately enhance your vision and logics to think critically and analytically.

## *Guidelines*

1. In order to attempt this assignment you should have full command on Lecture #5-9 and Lecture # 15-16
2. In order to solve this assignment you have strong concepts about following topics
   - ✓ Recurrence Relations and growth rate of the functions
   - ✓ Radix sort
3. Normally these formulas are very handy:

If $x^y = z$ then $y = \log_x z$

$a^{\log_b n} = n^{\log_b a}$

Also

$\sum_{i=1}^{n} a_i = \frac{n}{2}(a_1 + a_n)$ $\sum_{i=1}^{n} i = \frac{n}{2}(n+1)$ $\sum_{k=0}^{m} r^k = \frac{1-r^{m+1}}{1-r}$

$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ ( for n >=1)

Some basic information for solving assignment question 1 is given below.

**Growth Rate of Function:**

If some function $f_1(n) > f_2(n)$ for positive values of x then the function $f_1(x)$ is said to have greater growth rate then $f_2(x)$. For example $f_1(n)=n^{100}$ and $f_2(n)= n^{99}$ it is obvious that $f_1(x)$ has greater growth rate ( $2^{100} > 2^{99}$).This concept relate to complexity of algorithm ,an algorithm having greater growth rate function means the algorithm has greater complexity here $f_1(x)$ is more complex then $f_2(x)$.

## ***Books to read for solution***

Cormen, Leiserson, Rivest, and Stein (CLRS) 2001, **Introduction to Algorithms**, (2nd ed.) McGraw Hill.

## ***Estimated Time 3.5 hours***

Your concepts and logics will take actual measure of time ;however first question should not take more than 2 hour and for question 2 you may solve in 1.5 hours It all depends upon your sheer concentration.

## ***Question# 1*** **(10)**

Arrange the following in the **Least to Most** complexity order. Here "n "is the input size for the some complexity function and j< k and j & k are numbers greater than 2.Every function is separated by "comma" and note these are 20 functions to arrange.

$n/10000,\ 10n^{6k/2},\ n^{12j/4},\ \sqrt{n}\lg n,\ n^n,\ 1000000000,\ 2^n,\ \sqrt[7]{n}\lg n,\ n!,\ (2^n n\sqrt{n})/\sqrt{n^2},$

$n!/\sqrt{n},\ 2^n n\log n/\sqrt{n^2},\ n!/\log n,\ 1000000,\ n/\sqrt[4]{n},\ n(\log n)\sqrt[9]{n},$

$n/\sqrt[11]{n},\ n(\log n)\sqrt[3]{n},\ k^n,\ j^n$

## ***Question# 2*** **(10)**

Carry out the radix sort on the following five digits numbers and also develop complexities function and then write worst case Theta $\Theta$ notation for the radix sort algorithm.

45141,16545,11478,12196,12133,21322,31422,31511,11262,27210