CSN-212: Design and Analysis of Algorithms
(Spring 2015-2016)

Coding Project -1 (CP-1) Duration: Feb 1 – Feb 8, 2016

➢ Instructions:

- Objective of this Project:

  To know implementation of different sorting algorithms.

- Tasks to do in CP-1:

  1. Write the code in **C or C++ only** (**no other programming language; code from scratch without using any STL functions**) for the following list of sorting algorithms in separate files (name the files according to the algorithm name):
     i. Insertion sort
     ii. Selection sort
     iii. Bubble sort
     iv. Shell sort
     v. Merge sort
     vi. Quick sort
     vii. Heap sort
     viii. Counting sort
     ix. Radix sort
     x. Bucket sort

  2. Write the main() function in a top-level file <**filename.c**> or <**filename.cpp**> (a wrapper code) to take N (total number of elements to be sorted) from the user. Then generate N random elements (**integers only**) to be sorted as input data for any sorting algorithm. Store the input data in an 'array' (contiguous memory locations) and in a 'linked-list' (using dynamic memory allocation through malloc() or calloc()). Then call different sorting algorithms from main() with the input data-structure as argument. List the sorted elements as output.

  3. **Your codes for each sorting algorithms should be capable of handling both 'array' and 'linked-list' data-structures. You can write separate code files with inputs in two different data structures for each algorithm.**

  4. While writing a file for your code, you **MUST INCLUDE** the following information (for example) in the file before coding the actual program:
     ## GroupID-1 (14114XXX_14114YYY) - Name1 & Name2
     ## Date: February 1, 2016
     ## <filename>.c - State which algorithm is coded in this file.
     …
     Here you start writing your code.
     …

  5. Run your code for two extreme cases: (a) small size input data (e.g., 10 numbers) and (b) larger input data (e.g., 5000 numbers). Display the input and output list on the screen for the first case (a). Calculate the CPU time within your code (**excluding the random-number generation part**). Then report the CPU times for both the cases for all the sorting algorithms.

  6. Take the screenshot (by PrintScreen key) of one such run for each of the algorithms using both the data-structures. Hence, you will have in total 20 screenshot images in your report.

- Submission Method (Strictly Follow These):

  1. You must submit a zipped folder (<filename>.zip or <filename>.tar.gz) containing the following items:
     (a) All the code files named properly as mentioned above.
     (b) A report file (<filename>.DOC or <filename>.PDF) should contain your details like Group-ID, Name(s) and Enrollment Number(s) of the group member(s). The report should contain the following:
        i. The very brief descriptions of the algorithms or codes.
        ii. The codes you have written for the top-level file <filename.c> or <filename.cpp> (the wrapper code).
        iii. Take the screenshots as described above.

  2. Very Important: If the enrollment numbers of two members of your group are 14114XXX and 14114YYY, then replace <filename> with "14114XXX_14114YYY". Strictly follow this convention of you filename while submitting. If you do not follow this conventions, you (the group) will be given ZERO for CP-1.

  3. Submit your zipped folder (<filename>.zip or <filename>.tar.gz) through your account in Moodle. We have created a submission link in Moodle course site.

  4. Hard (Strict) deadline for submission: February 8, 2016 (8:00 am Indian Time).

  5. For any submission after Final Deadline, 5 marks will be deducted for every 24 hours of extra time.

  6. The key to success is starting early. You can always take a break, if you finish early.

- Evaluation Process:

  1. We will check the group formation information from the data received from you via the following link:
     https://docs.google.com/forms/d/1OeyAtKnQpKnJTjZnjv60iupZB8kZULqW-NRCYdStLpU/alreadyresponded
     If any discrepancy is found, the responsible students will get ZERO for CP-1 without any discussion with the course instructor.

  2. We will simulate your codes contained in your zipped folder and will regenerate the results you had included in the report file.

  3. Your submission will be checked with others' submission to identify any copy case. If we detect that your code is a copy (partially or fully) of other's code, then the total marks obtained by one group will be divided by the number of groups sharing the same code and will be given to all those groups who have duplicities.

  4. You may be asked to demonstrate and explain your submission after the submission deadline.