Assignment 1: Coding an Algorithm

Learning Outcomes:

After completing this Assignment, you should able to:

1. ***Design an algorithm using raptor.***
2. ***Implement the algorithm using C programing language.***
3. ***Estimate the order of growth of the running time***

Organisation

Please attempt this lab individually.

Assignment 1 - 20% of Module CA

Continuous assessment is based on ongoing assessment throughout the module. Personal circumstances will be considered on a case by case basis with supporting evidence.

Assignment Submission:

This assignment is given on Tuesday 20th February with a due date of Tuesday 10th of April 2018 at 12pm.

You need to submit all your assignment related files as a zip file named: DT211- 1-FirstnameSurname.zip through Classroom page, you zip file should include:

1. Flowchart.pdf (description of the flowchart and the steps to solve the following problems)
2. the Flowchart (Raptor source file for each problem)
3. the Program written in C for solving the following problems
4. and any other files you want to be considered for marking.

Late submissions (*within 2 weeks of the submission date*) will be marked out of 50% by giving a valid reason, otherwise 0 will grade will be awarded.

Design an Algorithm to express the following:

1) *Farthest pair (in one dimension).* Write a program that, given an array a[] of *N* double values, finds a *farthest pair : two values whose difference is no smaller than the difference of any other pair (in absolute value)*. The running time of your program should be linear in the worst case.

(25 Marks)

2) Closest pair (in one dimension). Write a program that, given an array a[] of *N* double values, finds a *closest pair : two values whose difference is no greater than the difference of any other pair (in absolute value)*. The running time of your program should be linear in the worst case.

3) Implement a stack with a single queue so that each stack operations takes a linear number of queue operations. *Hint* : To delete an item, get all of the elements on the queue one at a time, and put them at the end, except for the last one which you should delete and return

(25 Marks)

4) Implement each of the following operations on an array so that the time it takes does not depend on the array's size n

(i) *Delete the ith element of an array (1 ≤ i ≤ n)*

(ii) *Delete the ith element of a sorted array (the remaining array has to stay sorted, of course).*

(25 Marks)

*** In raptor you can use an array to simulate stack and queues.