

CSC 2720: Data Structures

Lab 3

Instructor: Shiraj Pokharel

Due : Next Day of the Lab @ 11:00 PM ET

Answer the below questions. You may use whatever IDEs / editors you like, but you must submit your responses on iCollege as .java files.

Note that your program's output must **exactly** match the specs(design , style) given here for each problem to pass the instructor's test cases .

Design refers to how well your code is written (i.e. is it clear, efficient, and elegant), while *Style* refers to the readability of your code (commented, correct indentation, good variable names).

This week as well, you will just play games with Arrays. :)

Image rotation is a fundamental image-processing operation required to be performed in the areas of Computer Graphics , Computer Vision(Artificial Intelligence/Machine Learning) and Visual Analytics (Data Science).

In today's Lab we will explore on ways to do a clock-wise rotation of an example bit-map representation of an image *aka* A Matrix. Below is the matrix :

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Our objective is to write (a) function(s) to rotate the above matrix by 90-degrees clockwise which produces the matrix as below:

$$\begin{bmatrix} 13 & 9 & 5 & 1 \\ 14 & 10 & 6 & 2 \\ 15 & 11 & 7 & 3 \\ 16 & 12 & 8 & 4 \end{bmatrix}$$

We will solve the problem in two ways:-

- (1) [50 points] Implement the function by allocating a new $n \times n$ 2D array. Then write the rotation to it by writing the rows of the original matrix to the columns in the solution matrix such that they fit the solution requirement. Then copy the new matrix exactly the same to the original matrix so that you know for sure you have updated the original matrix to look modified.

- (2) [50 points] Implement the function by **not** allocating a new $n \times n$ 2D array.
Hint : Perform the rotation in a layer by layer fashion - meaning - different layers can be processed independently. Also within a layer, you can exchange groups of four elements at a time to perform the rotation.
Example : Send 1 to 4's location, 4 to 16's location, 16 to 13's location and 13 to 1's location. Please feel free to use `get()` and `set()` methods if they help.

As a comment in your .java file for Problem 2, write your observation on the different solutioning methods utilized today. This week onwards your observation in a course like Data Structures should speak in terms of Time and Space Complexity. Which solution does what in terms of time and space. Which is better, which is worse. Which solution would you prefer and why? Your observation should be no more than 5 lines :)