CS 3000: Algorithms & Data — Summer 1 '20 — Tim LaRock

Homework 4
Due Tuesday June 2nd at 11:59pm Boston time via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Tuesday June 2nd at 11:59pm Boston time via Gradescope. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class, is strictly forbidden.

Problem 1. *Floyd-Warshall in LaTeX*

In class, we briefly discussed the Floyd-Warshall algorithm for finding shortest paths in weighted, directed graphs that do not have negative cycles. In this problem, you will use LaTeX to typeset the recursive definition and dynamic programming pseudocode for the Floyd-Warshall algorithm. You can find the relevant definition and pseudcode on the the Wikipedia page. You should also consider watching this 15 minute video by Abdul Bari explaining the algorithm.

(a) Starting from the template below, write a recursive definition for shortest paths using the *cases* LaTeX environment.

Solution:

$$X(m,n) = \begin{cases} x(n), & \text{for } 0 \leq n \leq 1 \\ x(n-1), & \text{for } 0 \leq n \leq 1 \\ x(n-1), & \text{for } 0 \leq n \leq 1 \end{cases}$$

(b) Based on your reading of the Wikipedia page and/or the Abdul Bari video, write a few sentences informally explaining why the recursive definition solves the problem. You should at least define each of the functions and variables that appear in your definition. Going forward, you should do this every time you write a recursive definition, even if we do not explicitly ask you to do so, otherwise we have to guess what you meant! Note that in this class, a little bit too much explanation is almost always better than not enough.

Solution:

(c) Starting from the template below, write pseudocode for the dynamic programming algorithm to find shortest paths between all pairs of nodes in a weighted, directed graph using the *algorithm* environment.

Solution:

Algorithm 1: MyAlgorithm $X[x_1, \ldots, x_n]$

$m \leftarrow \frac{n}{2}$
For $k = 0, \ldots, m$
  If *some condition* **:**
    do one thing (replace filler with the relevant LaTeX)
    do a second thing, using Ω
  ElseIf *other condition* **:**
    do something else
    then another thing with a fraction $\frac{y}{x}$
  Else
    do one last thing
$m \leftarrow 42$
Return the result