

Homework 03 (Due: Wednesday, November 13, 2024, 11 : 59 : 00PM (Eastern Time))

CPSC 3120

Instructions

This assignment consists of 4 analytical problems and 3 programming problems. Your solutions to the analytical problems must be submitted, as one PDF **without spaces, tabs, parentheses, pound signs, or percent signs in the filename**, via Canvas. While handwritten (then scanned) solutions to the analytical problems are acceptable, you are strongly encouraged to typeset your solutions in L^AT_EX or a word processor with an equation editor. The legibility of your solutions is of great importance.

Programming Assignment

Your methods will be tested on `newton.computing.clemson.edu`, using `gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.1)` and be compiled for C++ 2017. To ensure proper execution, you should review the reports that will be sent back to you on Canvas.

You will submit `cpsc3120homework03part01.h`, `cpsc3120homework03part02.h`, `cpsc3120homework03part01` and `cpsc3120homework03part02.cpp`, along with your PDF, via Canvas.

`assemblyLine`

`assemblyLine` is a function that takes four `vector<int>` arguments and returns an `int`.

A factory produces an item in n stages. The factory has two parallel assembly lines, each with n stations. The i^{th} station on Assembly Line 1 takes $s_{1,i}$ seconds to complete its task. The i^{th} station on Assembly Line 2 takes $s_{2,i}$ seconds to complete its task. It takes no time for an item to move between Station i and Station $i + 1$ on the same assembly line. However, when an item needs to be rushed, it can be moved between Assembly Line 1 and Assembly Line 2 as needed. The needed to move an in-production item from Assembly Line 1 to Assembly Line 2 after Station i and before Station $i + 1$ is $t_{AB,i+1}$ seconds.

Given a vector of times to complete each station on Assembly Line 1, a vector of times to complete each station on Assembly Line 2, a vector of times to transfer from Assembly Line 1 to Assembly Line 2, and a vector of times to transfer from Assembly Line 2 to Assembly Line 1 return the minimum time required to completely create an item in the factory.

`matrixChainMultiplication`

`matrixChainMultiplication` is a function that takes an $n \times 2$ `vector< vector<int> >` and returns an `int`. Given the row and column dimensions of n matrices, return the fewest number of multiplications required to generate the product of all of the matrices.

General Guidelines

Sample header, source, and testing files have been provided. You may modify the `.h` and `.cpp` files as needed, but you will only be turning in the four files mentioned above. The grading system will be compiling the code with the command

`g++ -std=c++17 -o /path/to/executable.out /path/to/source/files/*.cpp` for each part.

Written Assignment

Question 1 (10 points)

Question R-12.3 in *Algorithm Design and Applications*

Question 2 (10 points)

Question C-12.1 in *Algorithm Design and Applications*

Question 3 (10 points)

Question A-12.4 in *Algorithm Design and Applications*

Question 4 (10 points)

Question A-12.10 in *Algorithm Design and Applications*

Automated Report Notes

Reports will be generated every 3 minutes. Your programs should terminate within 60 seconds.

Point Allocation

Question	Points
Question 1	10%
Question 2	10%
Question 3	10%
Question 4	10%
assemblyLine	
Test Cases	1×20
Compilation	10
assemblyLine Total	30
matrixChainMultiplication	
Test Cases	1×20
Compilation	10
matrixChainMultiplication Total	30
Total	100%