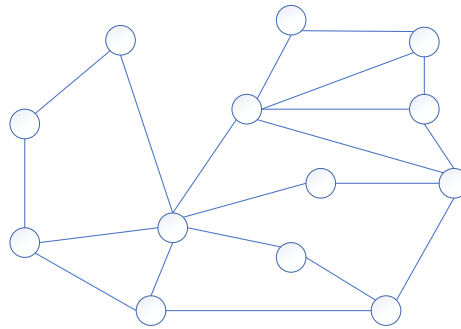


Computational Thinking and Problem Solving (COMP1002)

Assignment 3

(Due on **30 November 2022 (Wed) at 12:00 noon**)

1. [10 marks] Consider the following graph. Can the nodes be colored that neighbouring nodes do not share the same color? Explain briefly.



2. [20 marks] Refer to the Vending Machine example in Lecture 8 Slides 13. Modify the state diagram so that it also depicts the state transitions mentioned on Slide 14. That is, the vending machine should also be able to accept a \$2 coin and provide coin changing. Note that the user is able to abort after coins are inserted to the machine and the coins have to be returned to the user. Also, assume that the candy is automatically dispatched when sufficient values of coins have been inserted.

In your diagram, state clearly the meaning of each state and state transition.

3. [20 marks] Blockchain is a technology that maintains the transaction history of data and ensures it is immutable. A transaction in blockchain is defined as a change of the state of the blockchain. In other words, a transaction modifies the variables stored in the “database” of the blockchain. Also, a blockchain network consists of a number of nodes, which are essentially computers. Each of them stores a copy of the transaction history and all variables, collectively named as a Ledger. There are numerous implementations of blockchain. [Hyperledger Fabric](#) is one of them.

In Hyperledger Fabric, when an application, e.g., a Web-based warehouse management system, submits a transaction request, called transaction proposal, to the blockchain network, one of the blockchain nodes, called Endorser, is responsible to execute the transaction and create a digital signature on the transaction result. The signed transaction result is then passed to another node in the network, called Orderer, which is responsible for bundling a number of transactions into a “block” and broadcasting this block to other nodes, called Committer, and the Endorser in the network. Finally, the nodes receiving the new block will validate the transactions stored inside the block and update its Ledger with this new block. A transaction is said to be valid if it is signed by the Endorser and the transaction data has no conflict with the data currently stored in its Ledger.

Note that any Committer or Endorser in the network can always communicate with each other.

Answer the following questions:

- a) Draw a graph to depict the relationships of different nodes of the Hyperledger Fabric network. [10 marks]
- b) Depict a transaction in terms of a state diagram. [10 marks]

4. [15 marks] Define your own intersection function ($A \cap B$) in Python named `myIntersection(A, B)`, which finds a list of values from the intersection of two groups of data (A, B). The resulting list should contain no duplicated values. A and B can be a list or a tuple. Any built-in or external functions for intersection are not allowed.

You need to include a main logic to test the function in your .py file. An example is shown below:

```
A = [1,2,'a','b',3,'Hi',2,'A']
B = (3,2,'a','h',1,1,'abc','a')
print(myIntersection(A, B))
```

The print result is `[1, 2, 3, 'a']`. The order of the items in the list is not important.

Include your name and student ID as comment at the beginning of the file.

5. [35 marks] Write a Python program that fulfils the following requirements:

- a) Define your own *median* function in Python named `myMedian(X)`, which finds the *median* of a list, X, of at least 2 numeric values. [15 marks]
- b) Download `iris2021.csv` from Blackboard. In the csv file, the first row is the header, and the rest of the rows are the values. Write the main logic to calculate the median using `myMedian(X)` in a) for the column `Petal.Width` with respect to each type of *Species* in reverse alphabetic order. Your program MUST look like below:

```
The median of Petal.Width for virginica is 3.
The median of Petal.Width for versicolour is 3.
The median of Petal.Width for setosa is 1.
```

Note: Some parts of the output are masked intentionally.

The numbers are rounded to three decimal places.

Except the functions directly related to finding median, you are allowed to use built-in functions, but not any external modules/packages. Use a single .py file to include the code of a) and b) above. It is assumed that your csv file and .py file are put in the same folder.

Submission Instructions

Follow the steps below:

1. Create a folder and name it as <student no>_<your name>, e.g., **12345678d_CHANTaiMan**
2. For Q1, Q2 and Q3, type your answers in a word document and save it as a **.pdf** file. Name the single **.pdf** file as A3_<student no>_<your name>.**.pdf**, e.g., **A1_12345678d_CHANTaiMan.pdf**
3. For Q4 and Q5, submit the source file (**.py**). Name the **.py** files as A3_Q<question no>_<student no>_<your name>.**.py**, e.g., **A3_Q4_12345678d_CHANTaiMan.py**
4. Put all the **.pdf** and **.py** files into the folder created in Step 1.
5. Compress the folder (**.zip**, **.7z**, or **.rar**).
6. Submit the file to Blackboard.

A maximum of **3 attempts** for submission are allowed. **Only the last attempt will be graded.** A late penalty of 5% per hour will be imposed.

Any wrong file naming and submission will be given ZERO mark. It is your obligation to check carefully the files in your submission.

If you are using Windows, the file extension may be hidden by the operating system. Follow the steps of below links to make sure the file extension is not hidden:

<https://www.howtohaven.com/system/show-file-extensions-in-windows-explorer.shtml>

If your program cannot be run successfully (i.e., having any syntax error(s)) when it is triggered, ZERO mark will be awarded for that program, regardless of how much you have coded.

This assignment is an individual work. All work must be done on your own. Plagiarism is serious offence. You are not allowed to consult any external channels, e.g., discussion forums, and copy code from any web resources, to assist your completion of your assignments. The Moss (<https://theory.stanford.edu/~aiken/moss/>) system will be adopted for plagiarism checking for program code. Submissions with high similarity, in terms of code patterns and structures, in addition to direct-copy-and-paste, will be extracted and reviewed. Any plagiarism cases (both copier and copier) will be given ZERO mark plus a deduction of the maximum mark of this assignment. Serious cases would be submitted to the Student Discipline Task Group (SDTG) of the department for further disciplinary actions.