PRACTICAL EXAM – CSD203 – Spring 2024

Duration: 60 minutes

**Requirements:**

* Students are allowed to use all learning materials such as textbooks, slides, and lab exercises.
* Students are NOT allowed to use the Internet during the exam.
* Students are allowed to use IDEs such as PyCharm, Visual Studio Code, etc.

However, AI-assisted extensions such as TabNine or Copilot are prohibited.

* Students are NOT allowed to submit any personal information in the exam file such as name, student number, etc.
* Students must compress all of their work into a file named CSD203\_PE.zip.

**The exam files include this document, 2 python files named linkedlist.py and graph.py.**

**Question 1:**  
Use the **deque.py** file. Complete the following requirements:

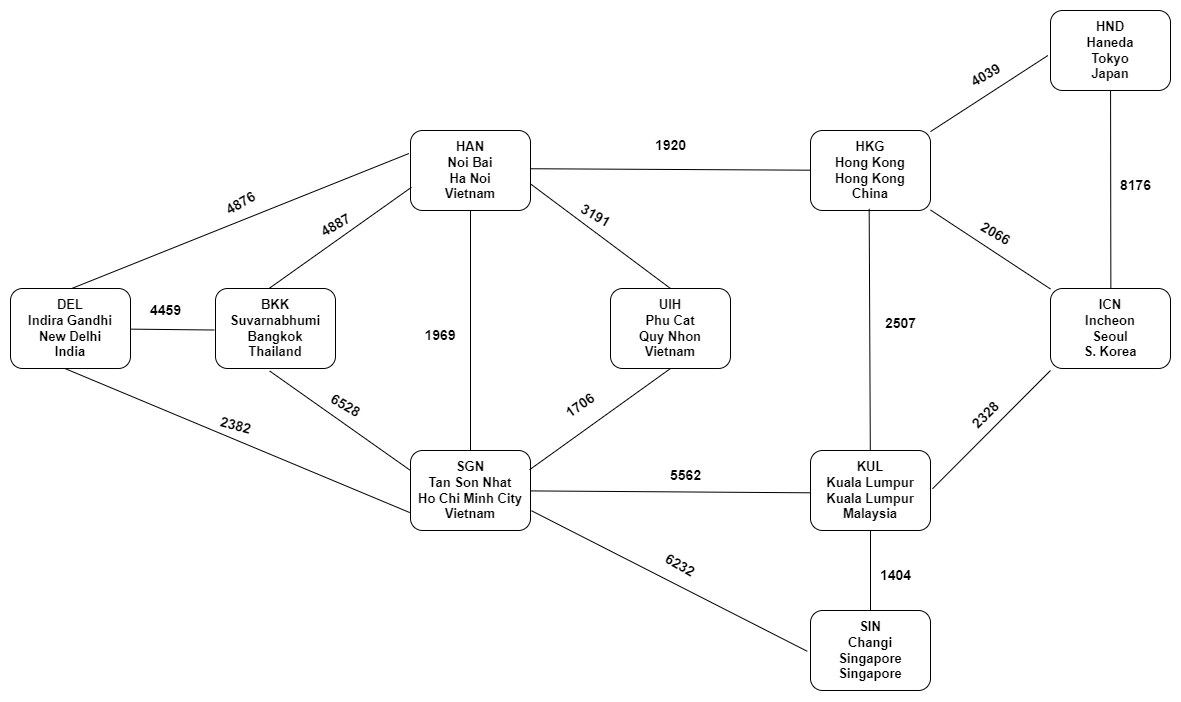
* Create an ***Airport*** class that includes the following properties: airport code, name, city, and country. The name, city, and country are default to None. (1.0 pt.)
* Complete the following methods of the ***AirportDeque*** class:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Methods | Descriptions | Marks |
| 1 | add\_first | Add an airport to the front of the deque. | 0.5 pts. |
| 2 | add\_last | Add an airport to the back of the deque. | 0.5 pts. |
| 3 | delete\_first | Remove and return the first airport from the deque; an error occurs if the deque is empty. | 0.5 pts. |
| 4 | delete\_last | Remove and return the last airport from the deque; an error occurs if the deque is empty. | 0.5 pts. |
| 5 | search\_airport | Finds all airports in the deque whose names contains the keywords. | 1.0 pt. |
| 6 | display | Show all the airports by their codes in the deque. | 1.0 pt. |

**Question 2:**

Use the **graph.py** file. Complete the following requirements:

* Implement the ***insert\_airport, insert\_flight*** methods and create a graph as the image below. (2.0 pts.)



(Nodes contain codes, names, cities, and countries)

* Implement the ***bfs*** method to traverse through the graph from an airport. The output should be a dictionary.

If we started from UIH, which cities we can visit after a direct flight / 1 transit / 2 transits, etc.

(1.5 pts.)

* Implement the ***shortest\_path*** method to find the most economic route between two airports in the graph. The output should be a deque of airports on the most economic route between those two airports, or None if two airports are not connected.

Find most economic route to travel from UIH to ICN.

(1.5 pts.)

**END**