

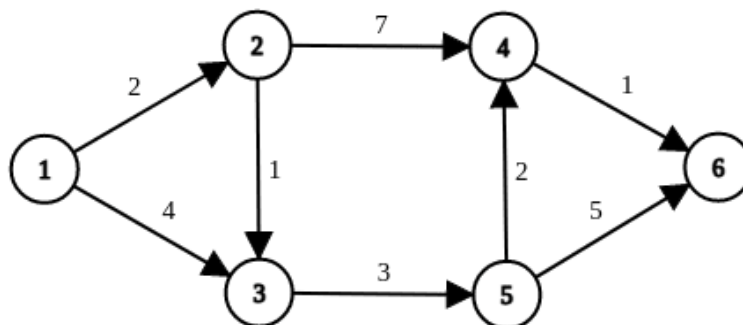
Coding Assignment: 2

Theme: Graph Algorithms

Rules:

- Use **Python 3.7** or higher in **Google Colab**. It comes with pre-installed packages like numpy, time, etc.
 - Submission should be a single **.zip** file (no other extensions like .rar, .gz) Keep this in mind.
 - Submission should contain
 1. Code files in **.ipynb** format. Use separate **.ipynb** files for separate questions. **DO NOT** use .py files. We will test your code in Google Colab.
 2. README file describing how to run the code. TAs will run your code to see if they are really working.
 3. A write-up in **.pdf** format. It should contain answers to the questions with appropriate screenshots or test results from the output of your code. Treat this as a mini-report that TAs will evaluate after your code successfully runs.
 - **DO NOT** use built-in functions for cases where you have been asked to implement that exact same functionality. You can use built-in functions if you want to test whether your designed function produces the correct output.
-

Q1. Dijkstra's Algorithm on Directed Graphs:



Implement Dijkstra's shortest path algorithm for the graph shown in the image above. Consider **node 1** as the source node. Calculate shortest paths and the weights of the corresponding

shortest paths for all the nodes, considering the source node as 1. You may take help from the tutorial code on Dijkstra's algorithm.

Q2. Bellman Ford's Algorithm for Shortest Path:

- a. We know that for negative edge weights in a graph Dijkstra's algorithm fails to find the shortest path. Change only one of the edge weights in the above graph of Q1 to a negative weight and implement **Bellman Ford's** algorithm to find out the shortest path to all nodes taking node 1 as the source node. For example: change the weight between **node 3** and **node 5** to -3 from 3.

Note: Don't change random weights to negative. Find out one such edge where a negative weight will force Dijkstra's Algorithm to fail.

- b. When does Bellman Ford's algorithm fail with negative weights? Show that using your code. Make necessary changes to the graph for demonstrating that.