Challenge 9: Remove Edge

In this challenge, we will learn how to delete an edge between two vertices.

We'll cover the following

- Problem statement
  - Input
  - Output
  - Sample input
  - Sample output
- Coding exercise

Problem statement#

You must implement the `remove_edge` function which takes a source and a destination as arguments. If an edge exists between the two, it should be deleted.

Input#

A directed graph, a source (integer), and a destination (integer).

Output#

A directed graph with the edge between the source and the destination removed.

Sample input#

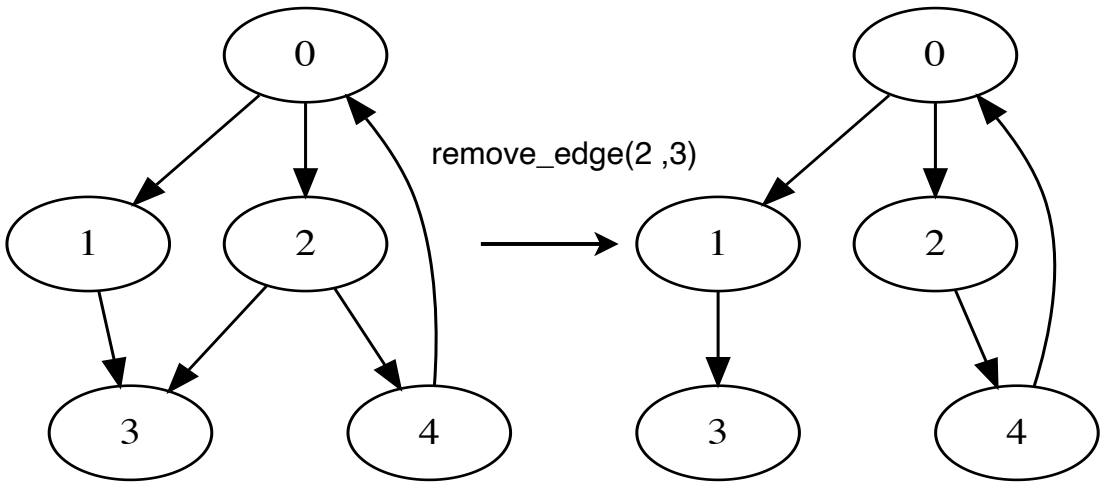
Vertex	Edges
0	1, 2
1	3
2	3, 4
3	None
4	0

Sample output#

```
remove_edge(graph, 2, 3)
```

Vertex	Edges
0	1, 2
1	3
2	4
3	None

Vertex	Edges
4	0



Coding exercise#

Take some time to flesh out the logic of your algorithm before moving on to the implementation. You have the previously implemented `LinkedList` class functions available for use.

Good luck!

main.py

Graph.py

Queue.py

Stack.py

Node.py

LinkedList.py

```
from Graph import Graph
from Queue import MyQueue
from Stack import MyStack
# You can check the input graph in console tab

# Create Stack => stack = MyStack()
# Functions of Stack => push(int), pop(), top(), is_empty()
# Create Queue => queue = MyQueue()
# Functions of Queue => enqueue(int), dequeue(), size(), front(), is_empty()
# class Graph => {int vertices, linkedList[] array}
# class linkedList => {Node head_node}
# class Node => {int data, Node next_element}


def remove_edge(graph, source, dest):
    # Write code here!
    return graph
```

null

Report an Issue