Challenge 6: Check if a Path Exists Between Two Vertices

Given a directed graph and two vertices, can you write a code to check if a path exists between the two vertices?

We'll cover the following

Problem statement#

You have to implement the `check_path()` function. It takes a source vertex and a destination vertex and tells us whether or not a path exists between the two.

Input#

A directed graph, a source value, and a destination value.

Output#

Returns `True` if a path exists from the source to the destinati



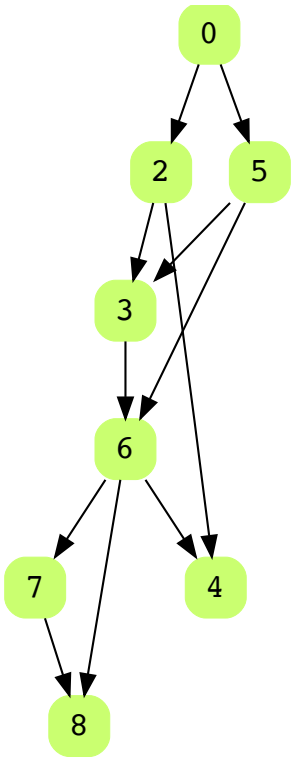
Sample input#

```
graph = {
    0 -> 2
    0 -> 5
    2 -> 3
    2 -> 4
    5 -> 3
    5 -> 6
    3 -> 6
    6 -> 7
    6 -> 8
    6 -> 4
    7 -> 8
}

source = 0
destination = 7
```

Sample output#

```
True
```



Path from 0 to 7:

0->5->6->7

0->2->3->6->7

0->5->3->6->7

Coding exercise#

Take a close look and design a step-by-step algorithm first before jumping to the implementation. This problem is designed for your practice, so try to solve it on your own first.

If you get stuck, you can always refer to the solution provided in the solution section. We will discuss the solution in the next lesson.

Good Luck!

main.py

Graph.py

Stack.py

Queue.py

LinkedList.py

Node.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 check_path(g, source, destination):
    # Write your code here
    pass

# Make helper functions here
```