# Challenge 5: Count Number of Edges in an Undirected Graph

In this lesson, we will figure out if it's possible to count the total number of edges in an undirected graph.

**We'll cover the following**



## Problem statement

You have to implement the `num_edges()` function which takes an undirected graph and computes the total number of bidirectional edges. An illustration is also provided for your understanding.

### Input

An undirected graph.

### Output

Returns the number of unique edges in the graph.

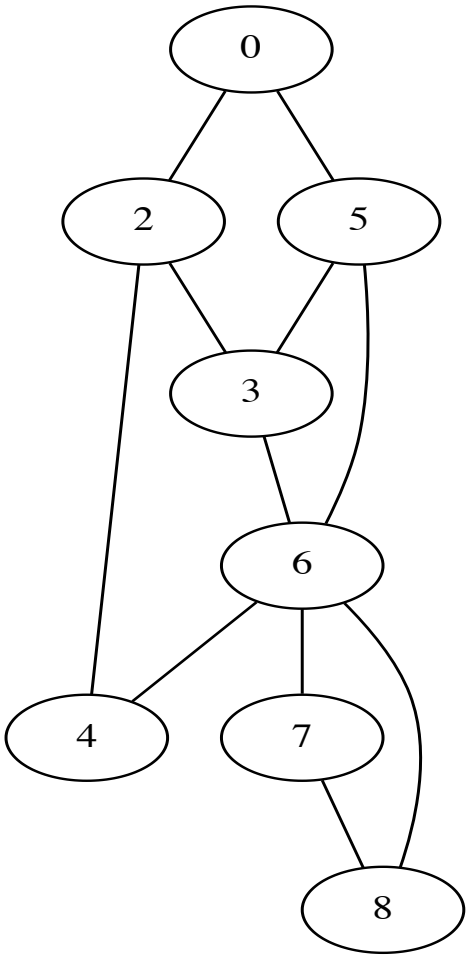


# Sample input

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

# Sample output

```
11
```




**Num of Edges : 11**

In the above diagram, even though the vertices have single edges between them the graph is still bidirectional.

# Coding exercise

This exercise is simpler than the previous ones. Nevertheless, take your time to solidify your logic before starting the implementation. Create as many helper functions as you need.

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 num_edges(g):
    # Write - Your - Code - Here
    pass
```



Back

Solution Review: Find a "Mother Verte...

Next

Solution Review: Count the Number of...

Mark as Complete