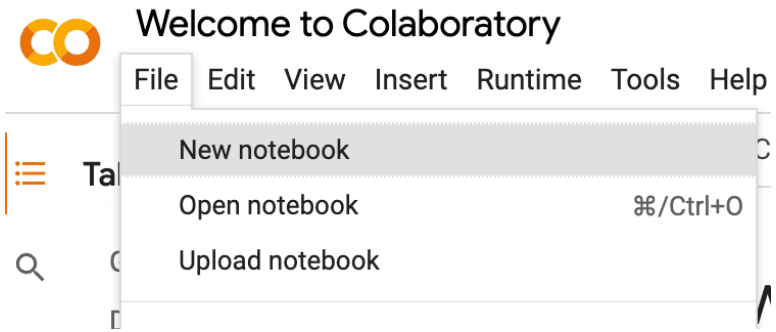
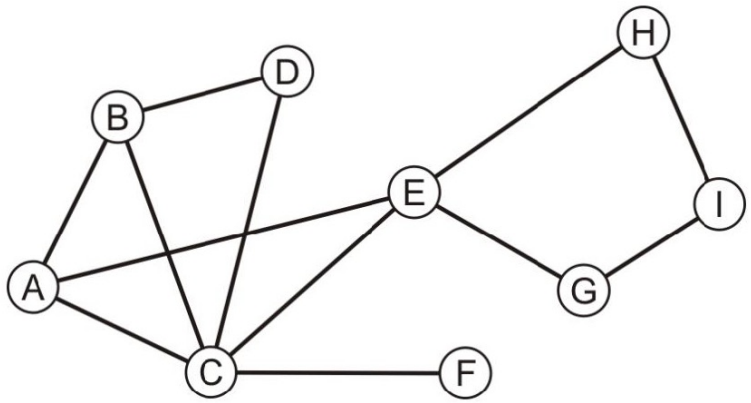
COMP9312 Python setup and Basic Graph Traversal

Setup

- Environment
  - Google colab
  - Jupyter Notebook
  - Python3
  - ….



Overview




```python
class SimpleGraph(object):
    def __init__(self, edge_list):
        self.vertex_dict = {}
        self.adj_list = []
        self.vertex_num = 0
        for [src, dst] in edge_list:
            self.add_edge(src, dst)

    def add_vertex(self, name):
        id = self.vertex_num
        self.vertex_dict[name] = id
        self.vertex_num += 1
        self.adj_list.append(set())

    def add_edge(self, vertex1, vertex2):
        if vertex1 != vertex2:
            if vertex1 not in self.vertex_dict.keys():
                self.add_vertex(vertex1)
            if vertex2 not in self.vertex_dict.keys():
                self.add_vertex(vertex2)
            self.adj_list[self.vertex_dict[vertex1]].add(vertex2)
            self.adj_list[self.vertex_dict[vertex2]].add(vertex1)
```

```python
import collections

> class SimpleGraph(object):···

> class DirectedWeightedGraph(object):···

> def getAllDegrees(G):···

> def BFS(G, v):···

> def DFS(G, v):···

> def connectivity(G, u, v):···

> def sumIndegree(G, u):···

> if __name__ == "__main__":···
```

Exercise 1: getAllDegrees()

- Understand the implementation of `class SimpleGraph`
  Manually load the graph in Figure 1 using and output the degrees of all the vertices. Check the function `getAllDegrees(G)`.

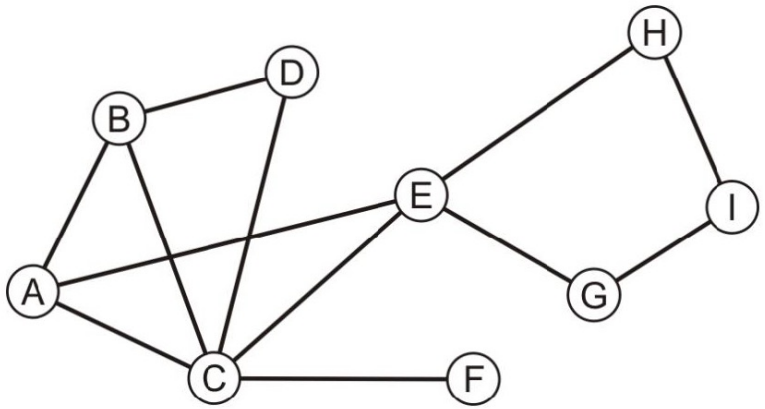
```
def getAllDegrees(G):
    # for each vertex of G, please print its degree here
    return
```

```
{'A': 0, 'B': 1, 'C': 2, 'E': 3, 'D': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
[{'B', 'C', 'E'}, {'A', 'D', 'C'}, {'A', 'D', 'F', 'B', 'E'}, {'A', 'G', 'C', 'H'}, {'B', 'C'}, {'C'}, {'I', 'E'}, {'I', 'E'}, {'G', 'H'}]
```

Now, you have 5 minutes to implement Q1:

Exercise 1: getAllDegrees()

```
def getAllDegrees(G):
  for u in G.vertex_dict.keys():
      print(u + " " + str(len(G.adj_list[G.vertex_dict[u]])))
```




```
A 3
B 3
C 5
E 4
D 2
F 1
G 2
H 2
I 2
```