CSE 4256 - Class 6

A graph is made up of two things: nodes (also called vertices) and edges. Shown below is a graph with 7 nodes and 9 edges.

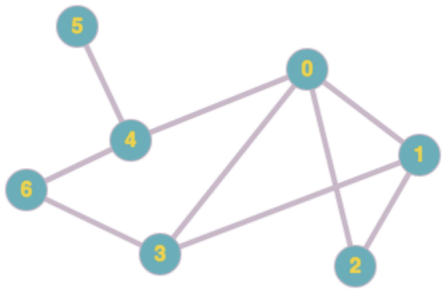

Figure 1: a simple graph

Graph Representations

We can represent this graph in multiple ways. One way is to use what is called an **adjacency matrix**. If a graph has n vertices, we create an n by n matrix m, and for each pair of vertices i and j, we let m[i][j] = 1 if there is an edge between vertex i and vertex j, 0 otherwise. Of course, we will represent a matrix by using a list-of-lists. The graph shown in figure one above is represented by the following matrix.

```
[[0, 1, 1, 1, 1, 0, 0],
 [1, 0, 1, 1, 0, 0, 0],
 [1, 1, 0, 0, 0, 0, 0],
 [1, 1, 0, 0, 0, 0, 1],
 [1, 0, 0, 0, 0, 1, 1],
 [0, 0, 0, 0, 1, 0, 0],
 [0, 0, 0, 1, 1, 0, 0]]}
```

Note the symmetry across the main diagonal.

We can also use an **adjacency-list** representation of a graph. We will use a dictionary to implement this representation. The keys will be integers and the values will be lists of integers.

```
d = {0: [1, 2, 3, 4],
     1: [0, 2, 3],
     2: [0,1],
```

```
    3: [0,1, 6],
    4: [0, 5, 6],
    5: [4],
    6: [3, 4]}
```

Finally, we can represent the graph using a list of edges:

```
[(0, 1), (0,2), (0, 3), (0,4), (1, 2), (1, 3), (3, 6), (4, 5), (4, 6)]
```

Note that, unlike with the previous representations, we don't include (i, j) in the list if (j, i) is already in it.

Working with Graphs in Python
You can loop over a dictionary using the items() method:

```
for key, value in d.items():
  #do something with the key and value
```

To see if a key is in the dictionary named d, use:

```
key_name in d
```

A 0-matrix of size nxn can be created using:

```
result = []
for i in range(n):
    result.append([0] * n)
```

To access the elements in a list named l, use:

```
for item in l:
    key = item[0]
    value = item[1]
```

Homework
In the exercises, you will practice converting one representation of a graph into another representation. The first problem has been solved for you.

Question 1
convert a dictionary representation of a graph to a matrix representation.

```
def dict_to_matrix(d):
    # Your code goes here.

d = {0: [1, 2],
1: [0, 2, 4],
```

```
2: [0, 1, 3],
3: [2, 4],
4: [1, 3]}

result = dict_to_matrix(d)

for row in result:
    print(row)

# result is
# [0, 1, 1, 0, 0]
# [1, 0, 1, 0, 1]
# [1, 1, 0, 1, 0]
# [0, 0, 1, 0, 1]
# [0, 1, 0, 1, 0]
```

Solution

```
def dict_to_matrix(d):
result = []
for i in range(len(d)):
  result.append([0] * len(d))

for key, value in d.items():
    for number in value:
      result[key][number] = 1


******************
```

Question 2

```
#convert a dictionary representation of a graph to a list representation.

def dict_to_list(d):
    # your code goes here.

d = {0: [1, 2],
1: [0, 2, 4],
2: [0, 1, 3],
3: [2, 4],
4: [1, 3]}

print(dict_to_list(d))

#result is
#[(0, 1), (0, 2), (1, 2), (1, 4), (2, 3), (3, 4)]
```

****

Question 3

```
#convert a list representation of a graph to a dictionary representation.

def list_to_dict(l):
    # Your code goes here.

l = [(0, 1), (0, 2), (1, 2), (1, 4), (2, 3), (3, 4)]

print(list_to_dict(l))

#result is
#d = {0: [1, 2],
# 1: [0, 2, 4],
# 2: [0, 1, 3],
# 3: [2, 4],
# 4: [1, 3]}
```

****

Question 4

```
#convert a list representation of a graph to a matrix representation.

def list_to_matrix(l):
    # Your code goes here.

l = [(0, 1), (0, 2), (1, 2), (1, 4), (2, 3), (3, 4)]

result = list_to_matrix(l)
for row in result:
    print(row)

#result is
# [0, 1, 1, 0, 0]
# [1, 0, 1, 0, 1]
# [1, 1, 0, 1, 0]
# [0, 0, 1, 0, 1]
# [0, 1, 0, 1, 0]
```

****

Question 5

```
#convert a matrix representation of a graph to a dictionary representation.
```

```
def matrix_to_dict(m):
    # Your code goes here.

m = [[0, 1, 1, 0, 0],
     [1, 0, 1, 0, 1],
     [1, 1, 0, 1, 0],
     [0, 0, 1, 0, 1],
     [0, 1, 0, 1, 0]]

print(matrix_to_dict(m))

#result is
# d = {0: [1, 2],
#  1: [0, 2, 4],
#  2: [0, 1, 3],
#  3: [2, 4],
#  4: [1, 3]}
```

****

Question 6

```
#convert a matrix representation of a graph to a list representation.

def matrix_to_list(m):
    # Your code goes here.

m = [[0, 1, 1, 0, 0],
     [1, 0, 1, 0, 1],
     [1, 1, 0, 1, 0],
     [0, 0, 1, 0, 1],
     [0, 1, 0, 1, 0]]

print(matrix_to_list(m))

#result is
#l = [(0, 1), (0, 2), (1, 2), (1, 4), (2, 3), (3, 4)]
```