Undirected graphs

Help Center

We will represent graphs as a dictionary utilizing nodes as keys and adjacency sets (i.e. the node's neighbors) as values. Therefore each node in the graph is a key in the dictionary that maps to a set containing that node's neighbors. For example the graph $G = (V, E)$ where $V = \{0, 1, 2\}$ and $E = \{\{0, 1\}, \{1, 2\}\}$ can be represented as:

```
graph = { 0 : set([1]),
          1 : set([0, 2]),
          2 : set([1]) }
```

Exercise 1 - Node Count

Write a function:

```
def node_count(graph):
    """
    Returns the number of nodes in a graph.

    Arguments:
    graph -- The given graph.

    Returns:
    The number of nodes in the given graph.
    """
    ...
```

that computes the number of nodes in a graph.

You can obtain a list of a given dictionary's keys by using the `keys()` method. For example:

```
graph = { ... }
nodes_in_graph = graph.keys() # A dict's keys define a graph's nodes!
```

Assert that your function produces the following output:

```
>>> graph1 = { 0: set(), 1: set()}
>>> node_count(graph1)
2
```

```
>>> graph2 = { }
>>> node_count(graph2)
0
```

Exercise 2 - Edge Count

Write a function:

```
def edge_count(graph):
    """
    Returns the number of edges in a graph.

    Arguments:
    graph -- The given graph.

    Returns:
    The number of edges in the given graph.
    """
    ...
```

that computes the number of edges in a graph.

Assert that your function produces the following output:

```
>>> graph1 = { "0" : set(["1","2"]),
               "1" : set(["0","2"]),
               "2" : set(["1","0"]) }
>>> edge_count(graph1)
3
```

```
>>> graph2 = { 0: set(), 1: set()}
>>> edge_count(graph2)
0
```

```
>>> graph3 = { }
>>> edge_count(graph3)
0
```

Created Sat 23 Aug 20149:17 PM BST

Last Modified Sun 24 Aug 2014 5:47 AM BST