Dijkstra's Algorithm

The purpose of this project is for you to write codes that implement Dijkstra's Algorithm. In addition, you will determine whether or not a given graph is connected. The ONLY IMPORT ALLOWED is 'copy', and you are ONLY allowed to use the copy.deepcopy() method from this package. All of your code should be 'from scratch.'

Objectives

You are to write a function `infty(graph)` that takes in a weighted graph as its input and returns the sum of all the edge-weights plus 1.

- `infty({"A" : [["B", 10], ["D", 5]], "B" : [["A", 10], ["C", 5]], "C" : [["B", 5], ["D", 15]], "D" : [["C", 15], ["A", 5]]})` should return 36.

You are to write a function `initial(graph)` that takes in a weighted graph as its input and returns a vertex-coloring in which "A" is colored with 0 and all other vertices are colored with 'infty'.

- `initial({"A" : [["B", 10], ["D", 5]], "B" : [["A", 10], ["C", 5]], "C" : [["B", 5], ["D", 15]], "D" : [["C", 15], ["A", 5]]})` should return {"A" : 0, "B" : 36, "C" : 36, "D" : 36}

You are to write a function `find_min(color, queue)` that takes a vertex-coloring and a list of vertices, and returns a vertex in the list, whose color is smallest.

- `find_min({"A" : 0, "B" : 10, "C" : 10, "D" : 15}, ["A", "D"])` should return "A"
- `find_min({"A" : 0, "B" : 10, "C" : 10, "D" : 15}, ["B", "C", "D"])` should return "B" or "C"

You are to write a function `dijkstra(graph)` that takes in a weighted graph as its input and returns the vertex-coloring outputed by Dijkstra's Algorithm where "A" is the source.

- `dijkstra({"A" : [["B", 10], ["D", 5]], "B" : [["A", 10], ["C", 5]], "C" : [["B", 5], ["D", 15]], "D" : [["C", 15], ["A", 5]]})` should return {"A" : 0, "B" : 10, "C" : 15, "D" : 5}
- `dijkstra({"A" : [["B", 10], ["D", 5]], "B" : [["A", 10], ["C", 5]], "C" : [["B", 5], ["D", 15], "D" : [["C", 15], ["A", 5]], "E" : [["F", 5]], "F" : [["E", 5]]})` should return {"A" : 0, "B" : 10, "C" : 15, "D" : 5, "E" : 41, "F" : 41}

You are to write a function `is_connected(graph)` that takes in a weighted graph as its input and determines whether or not the graph is connected.

- `is_connected({"A" : [["B", 10], ["D", 5]], "B" : [["A", 10], ["C", 5]], "C" : [["B", 5], ["D", 15]], "D" : [["C", 15], ["A", 5]]})` should return True.
- `is_connected({"A" : [["B", 5], ["C", 5]], "B" : [["A", 5], ["C", 5]], "C" : [["B", 5], ["A", 15]], "D" : [["E", 5]], "E" : [["D", 5]]})` should return False {"A" : 0, "B" : 10, "C" : 15, "D" : 5}

Grading Rubric

Your functions will be tested using a collection of pre-made test cases that I will create. Your grade will be based on how often your code produces correct results and on the quality of the descriptions that you provide for your functions. Full credit will be given to a notebook whose functions work 100% of the time. Also each function has a description that clearly explains how the function works.