



Please note the following regarding the task below:

- Language used should be Python
- Push your code to GitHub, and send us the link to the repository
- Lastly, do not hesitate to contact us for any questions :)

You are given a json file which represents an undirected graph G of persons with some qualities q (the definition of qualities does not matter, just know that the higher the value, the better). The *nodes* dictionary shows all nodes/vertices that exist, whereas the *links* dictionary shows how these nodes are connected (edges).

Here is what one node looks like:

```
{"id": "48633953191", "q1": 266, "q2": 0.496241}
```

And here is what a link looks like:

```
{"id": "48633953191|56010353", "source": "48633953191", "target": "56010353"}
```

Your task is to manually create the following graph metrics and tests as functions (note that you can use built-in *networkx* metrics to **test** your metrics.):

- Vertex and edge counts
- Vertex degree
- Test if the graph is connected
- Count and create the subgraphs (components) of G
- Count the isolates, and show them
- Eigenvector centrality
- Betweenness centrality
- **[Optional]** Implement hierarchical clustering solely based on the network structure (disregard node qualities) and show the resulting dendrogram. You can take inspiration from existing implementations. For the visualization of the dendrogram, use existing plotting tools – no need to implement this manually.

Lastly, based on the qualities q , and the two centrality metrics, rank the nodes (use any algorithm/approach you see fit). This is more of a creative approach, so there is no right or wrong way - just elaborate on your choice. Note that, if you choose an algorithm for this task, you do not have to manually implement it.

Good luck and have fun!