Significant Nodes
This project will be an application that identifies the most influential users for a particular hashtag on Twitter using graphs. To get the influence of particular nodes (Twitter users), I will implement page rank to get a list of important nodes. Then using Djikstra's algorithm, we will find the shortest paths between two selected nodes.

Competitive Analysis:
I have not seen any project directly tied to this as most of this type of work happens within a jupyter notebook. I propose a real-time implementation of this work such that even non-programmers can search for Twitter data and identify the leaders of an online conversation.

Structural Plan:
The following functionality: collecting tweets, structuring Twitter data, visualizing the graph, and applying algorithms (either Djikstra / Page Rank), will be worked on in separate files / modules. They will all later be imported into a main function from which the entire application can be run. I have also divided the app into interactive and terminal mode. In terminal mode, the app can be run without a UI. This is much better for performance and we can see the different graphs on the screen. **Currently,** the terminal mode is fully completed

Algorithmic Plan:
I will use an adjacency dictionary to represent the edges and nodes of a graph. To structure the data in this dictionary, I will make use of pandas to semi-process information collected from a CSV file generated from the Twitter API.

Format of the adjecency dictionary:

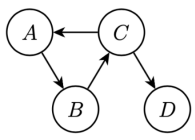
Adjacency dictionary

```
adj_dict = {"A": {"B":1.0}, "B":{"C":1.0}, "C":{"A":1.0, "D":1.0}, "D":{}}
```

This adjacency dictionary encodes the following facts:

`adj_list["A"] = {"B":1.0}` means that node "A" has a link to node "B" (e.g. with weight 1.0)
`adj_list["B"] = {"C":1.0}` means that node "B" has a link to node "C"
`adj_list["C"] = {"A":1.0, "D":1.0}` means that node "C" has links to nodes "A" and "D"
`adj_list["D"] = {}` means that node "D" has no outgoing links

Node (key)	Edges
A	{B:1.0}
B	{C:1.0}
C	{A:1.0,D:1.0}
D	{}



The hardest part of this will be implementing PageRank or Djikstra's algorithm from scratch. On speaking with the professor, we came to the conclusion that I should try to have one of these algorithms implemented from scratch by the time of my first deliverable. If developing the algorithms isn't feasible, then my focus should be on getting the UI and UX to a state of usability that is as nice as possible.

Version Control Plan:
I will use git for version control. My code will also be pushed onto a public GitHub repository for easy access.

Module List:
Pandas
Tkinter
Networkx
PyInquirer
Tweepy