Summary:

* Node = an individual faculty
* Edge = a similarity between a pair of faculties
  1. There are many ways to define and compute the weights of edges but we are going to compute the similarity of revealed research interests/identity between faculties based on research interests or descriptions in their faculty profile pages.
  2. Using pre-trained embedding may be the best approach, considering the limitations of other approaches (e.g., text matching and TF-IDF).
* Workflow:
  1. Text pre-processing to remove terms that are not meaningful or negatively affect the measurement of similarity (e.g., school names); Although I think this issue may be critical only for text matching approaches including TF-IDF, there may be some cases where I could not think of so please make sure to communicate with other group members whenever you find something concerning!
  2. Computing similarity scores using discussed techniques (e.g., TD-IDF, pre-trained embeddings, etc.); there might be better ways to do it so please share with everyone if you found something cool!
     + Check out Fuzzy String Matching (<https://towardsdatascience.com/fuzzy-string-matching-in-python-68f240d910fe>) and other ways of computing text similarities including pre-trained embeddings and TD-IDF (<https://medium.com/@adriensieg/text-similarities-da019229c894>)
  3. Create two tables: one with dyadic relationships, i.e., node 1, node 2, and weight of similarity between nodes 1 and 2 as a row; and another one with node attributes, i.e., node name in a column and important attributes that we may want to explore such as the division information in other columns.
  4. Constructing a network; some pointers are here:
     + <https://towardsdatascience.com/visualizing-networks-in-python-d70f4cbeb259>
     + <https://programminghistorian.org/en/lessons/exploring-and-analyzing-network-data-with-python>
     + <https://pyvis.readthedocs.io/en/latest/>
  5. Pruning network:
     + Backbone algorithm:
       - <https://f.briatte.org/r/disparity-filter-for-directed-networks> (quick summary)
       - <https://github.com/aekpalakorn/python-backbone-network> (Python)
       - <https://rdrr.io/cran/disparityfilter/man/backbone.html> (R)
     + One very simple but still reasonable and scientifically sounding approach: setting up a threshold for a similarity score (i.e., removing those edges with lower similarity scores)
  6. Advanced network visualization with clusters
     + You may want to search for not only “network clusters” but also “community detection” because both can be considered more or less the same thing in this type of network analysis.
     + Some pointers:
       - <https://towardsdatascience.com/community-detection-algorithms-9bd8951e7dae>
       - <https://orbifold.net/default/community-detection-using-networkx/>
       - <https://pyvis.readthedocs.io/en/latest/>
       - <https://melaniewalsh.github.io/Intro-Cultural-Analytics/06-Network-Analysis/02-Making-Network-Viz-with-Bokeh.html>
       - You can simply do one very popular one called Louvain with Python’s networkx: <https://medium.com/analytics-vidhya/implement-louvain-community-detection-algorithm-using-python-and-gephi-with-visualization-871250fb2f25> (this example shows how to do it with Gephi, which I mentioned earlier as a tool that you may want to use for beautifying a network visualization).
       - There should be a ton of tutorials. So, please find other ones if these are not useful.
  7. Furthermore:
     + Network measures: <https://datascience103579984.wordpress.com/2019/09/12/26-network-analysis-in-python-part-1-from-datacamp/>
     + Topics of faculties (e.g., topic modeling?)
     + And more