A1: Can you summarize the state of the art you learned in no more than one page?

This paper seeks to explore the hidden social hierarchies and the role of prestige in faculty hiring among 206 institutions across three different domains. The paper takes advantage of statistical techniques, such as the Kolmogorov-Smirnov (KS) test, Markov Chain Monte Carlo (MCMC) optimization, and the Gini Coefficient and Lorenz curve, to support its findings. This paper makes use of the minimum violating rankings method to devise a ranking system for the institutions across the three domains, thus making prestige an operational variable. Furthermore, it tries to understand how prestige affects non-meritocratic factors like social status, geography, etc.

The KS test is a nonparametric test that determines whether given cumulative distribution functions originate from the same underlying distribution. This probabilistic test quantifies the probability of the null hypothesis by returning an upper bound. The null hypothesis essentially states that the empirical distributions have been generated from a common underlying distribution. To denote the probability of this being true, we use probability P, and the KS test then determines an upper bound for P. Therefore, a small value of the upper bound for P indicates that with a high probability that the empirical CDFs are statistically different from the underlying distribution. The paper employs this test to reject the size-proportional-placement hypothesis which claims that placements are proportional to the size of academic units within institutions. This hypothesis implied indistinguishable distributions of both placement and size which would have supported the proportionality claim. However, the KS test suggested otherwise, as it returned an upper bound of $P < 10^{-8}$ indicating that these distributions are not alike since there is a high chance that they're not sampled from the same underlying distribution. This application of the KS test helps establish that faculty placement is independent of the size of academic units.

The Lorenz Curve and Gini Coefficient are used to quantify income inequality for a population against a standard known as the line of equality. The Lorenz Curve represents the percentage of wealth held vs the cumulative percentage of the population where curves below the line of equality imply that a small portion of the population holds most of the population's wealth. The Gini Coefficient then gives a ratio of the intersected area between the Lorenz Curve and the Line of Equality with the total area under the Line of Equality (value lies between 0 and 1 mostly). This paper graphs a variant of the Lorenz Curve where one of the axes is the fraction of institutions instead of the cumulative percentage of institutions. This produces a graph that is essentially a reflection of a normal Lorenz Curve across the y = x-axis. The paper finds G to be between 0.62 and 0.76, which is higher than America's current Gini Index of 0.41. This implies that only a small fraction of institutions produce the most faculty that go to other institutions. Additionally, this sheds light on whether faculty hiring is solely determined by meritocratic factors like doctoral prestige, or if other non-meritocratic factors also have an impact. The paper asserts that for faculty hiring to be purely meritocratic, the top 10 must have a production that is 2 to 6 times better than the third ten. The paper suggests this claim is implausible, as the supporting data suggests there are other factors that come into play.

The third method devises a ranking for the nodes in the network. This method shares the same purpose with other algorithms like PageRank but has a different approach. Since the network is a huge directed acyclic graph (the dataset has self-loops, but they have been omitted since they're trivial when you implement the method) there is a need to rank the nodes to pinpoint nodes that are "central"/ "important". The Minimum Violation Rankings algorithm has a simple goal - minimize the edges going up, i.e., maximize the edges going down. Why such an optimization problem? To devise an optimal rank permutation π, we need to ensure that the network is closest to a perfect social hierarchy - where there is less movement up the triangle and more movement down. This implies that there should mostly be positive changes in π as you go from Node u to Node v (since a lower value means more prestige). One drawback with this method is that there can be multiple permutations that can minimize the "violations" in each network and the rank uncertainty can be very high due to the spread of the rank data. Hence, the (supplemental and main) paper suggests a method (based on MCMC optimization) to simplify this complex problem - choose a random pair (u,v) and compute and compare the current and new minimum violations ranking after you swap the ranks. If the new ranking is better or equal to (this can solely depend on whether it is a min or max problem) the current ranking, then update the current ranking and the current. By doing so, you can get some nodes that don't have their ranks swapped at all or those that have very small uncertainties associated with their ranks. (Note that, self-loops are omitted since they represent no movement in the hierarchy.)

A2: Can you run some interesting queries over the graph in Neo4j? Report 2-3 queries and what the results are.

I ran three queries - one is to load the CSV into the Neo4j browser and create the basic graph and then I ran two interesting queries to extract some useful information from the graphs. **Query 1** reads the input CSV file that I created using Datasets 3 & 4 and based on the category of Type, it assigns the directed edges accordingly. It then

initializes all the attributes associated with the nodes and edges and a small subset of the graph is displayed to the user. **Query 2** finds a variable length path of type "FACULTY" between a user-inputted starting and end node where all intermediary edges have the "Gender" attribute as "M", the "Rank" as "Assoc" and this resultant path is also the one that minimizes the sum of the rank changes as one traverses the path (smallest magnitude of the sum of all in the path). This query is interesting since the shortest path gives a realistic path for somebody who is trying to get hired at a particular institution starting from some other institution. This query also fixes a drawback of the shortest path which blatantly ignores the change in ranking in the path - by minimizing the change in rank and specifying other attributes, one can get an even better idea of how realistic it is to reach their end goal. **Query 3** returns the shortest variable lengths path from a starting node to other nodes that are within the same region and are ranked better than the starting node. This query is important because if somebody has a very good academic background and potential and is looking to work at high-ranked institutions, they don't have to burden themselves financially by moving far from where they started, rather they can work towards the shortest path towards the elite institutes within their region.

A3: Implement MVR. Do you get the same results as the paper?
I created the adjacency matrix for the network, and I omitted the "All Others" node since it was not an institution on its own and could have created discrepancies. I ran the MVR algorithm for 50,000 iterations which took approximately 2 hours, and I also returned the average of the rank (which were mostly whole numbers) as well as the uncertainty associated with each rank. I also created a plot to show the MVR value versus the number of iterations to check for convergence. I used standard deviation to determine the uncertainty for the ranks. In contrast to what the paper claimed, I observed the average rank uncertainty for the bottom 50 universities to be 5 times that of the top 50 universities. This was something I didn't expect since very low-ranked universities should share similar rank uncertainty characteristics with prestigious universities. Also, I used Figure S10 provided in the supplementary material to compare my results with the Top 60 institutions. I computed the mean and median for the rank difference between my results and these 60 institutions and I found that the skew was positive (right) where the mean was 6.25 and the median was 3. This means that there are some outliers that try to pull the mean up, but the median gives a good idea of what the typical rank difference is. Lastly, I only found 5 universities from the 60 (8.33 %) which shared the same rank as my MVR results. Overall, I feel like my results were not very far off from the paper.

A4: Do you think the "ranking" method suggested in the paper is good? Any issues with it? Do you suggest improvements?
It is a unique method of ranking because it is trying to create a perfect social hierarchy in a network by finding possible permutations for the ranks that maximize the number of edges going "down". For a researcher, the method is easy to replicate and understand, however, it has some practical limitations. Before delving into practical issues, we can discuss the theoretical ones - the method disregards other factors like the incoming/outcoming edges of a node, the average distance of a node from all other nodes, and the rank of a node's neighbors. Regarding the implementation, it is computationally very expensive as it travels through each edge and node for a lot of iterations. Also, it is hard to determine when to stop since the MVR value doesn't seem to converge for a higher number of iterations. An improvement would be to experiment with different initial/starting values for the ranks and include an epsilon ε to put an upper bound on the convergence of the MVR values.

A5: Implement the method. Compare the ranking results to Part 3. Are they different? Explain how and why they are different. Which one makes more sense to you?
The first 12 pages of this paper discuss popular centrality measures from which I used the method of PageRank. The rankings that I obtained from PageRank are dissimilar to the ones obtained from MVR. This is because PageRank determines the importance of each node by the incoming relationships (edges) that it has and considers the importance (through the PageRank score) of the "source" nodes – it believes that an end node's importance is solely determined by the source nodes that are connected to it. Furthermore, the algorithms have different implementations – PageRank uses Markov chains and the idea of a Random surfer (stochastic processes) whereas MVR uses a local greedy algorithm to solve a constrained optimization problem. MVR fails to consider the importance of neighboring nodes (unlike PageRank) and it also is much slower than PageRank since it visits all nodes/edges due to the lack of an established convergence threshold. To me, PageRank is a more sensible algorithm as it gives a better idea of the importance of a node, i.e., the PageRank score is determined by the quality of the connection and not the quantity (unlike MVR).