1. The fatal flaw with this algorithm is that the result will not be a perfect matching. There may be some advisors with no students after three rounds. The key difference between this algorithm and Gale Shapley algorithm is how the advisors give offers to students. In this algorithm, advisors always make at most three offers. In the Gale Shapley algorithm, one advisor would give an offer to a student, and is not limited with how many offers they can give.
2. To fix this fatal flaw, I will change the algorithm so there are multiple rounds of offers until all advisors get a student.
3. Proof by contradiction. Assume that using both GPA and unique location distances to decide the advisor’s preference does not give a total ordering. This means that student a and student b are tied. If they are tied, they have the same GPA and same location distance, but this contradicts our statement that each student has a unique location distance, therefore using both GPA and unique location distances to decide the advisor’s preference list still gives total ordering.
4. In the Gale-Shapley algorithm, the person who is proposing gets a better matching than if they were the one being proposed to. In our case, we can have the students “propose”, This way, they get the best possible outcome for themselves and should have no incentive to lie.
5. Each advisor makes a preference list, ranking students higher if they have a higher GPA. If there are ties, then the advisors break the tie by choosing the student closer to them. The students also make preference lists. We start the round. If the advisor does not have a student, they will pick the student who ranks higher on their preference list that they have not given an offer to before. Then the advisors give offers to their students. Each student then looks at their offers. If they have no offer, they don’t do anything. If they have multiple offers, they pick the one higher on their preference list. If they do not already have an advisor, they pick that advisor. If they do already have an advisor, they compare them with the one they picked and choose the advisor that ranks higher in their preference list. Now that advisor does not have a student. We go back to the beginning of the rounds if there is an advisor who still does not have a student. Otherwise, we are finished.
6. Copy from powerpoint of GS and adjust
7. Asdf
8. Using the brute force algorithm to get all the permutations of matchings takes O(n!). For each permutation, verifying whether they form a weakly stable marriage takes O(n^2). So the overall runtime complexity of the brute force algorithm with verification of stable marriage is O(n^2\*n!).
9. asdf