George Zhang
Computer Science 227r
Cynthia Dwork, Victor Balcer
Due: March 20, 2017

Final Project Idea

For my final project, I would like to implement a few of the algorithms we have learned during the semester. My motivation for this comes from my own learning style; I very much like seeing how things we learned are applied. The lecture I found most engaging thus far has been the last lecture before spring break in which we discussed Google's algorithm for finding the most popular "home page" websites in a differentially private way. In that lecture, I was able to see how the randomized response algorithm could be applied in a particular context.

With that in mind, I think the project I would most be interested in would be to create a set of toy examples for various algorithms that could be used in future iterations of the course. My hope would be that seeing the algorithms in a particular context, even if in a toy example, could appeal to students like me, thus broadening the audience for the course. However, I do have two concerns about this: I worry that this may fall under the "implement the algorithm and show that it works" umbrella, and I currently do not have a good idea how I would frame the algorithm in an engaging way (that is, asking a query and getting back a number is not particularly enlightening; how would I show that this is privacy preserving?). These two obstacles may prove to be a problem. One potential way to overcome them would be to address the question proposed in the spec: "How do multiple algorithms for the same problem compare for different types of inputs?" This would allow me to implement several algorithms and proceed with a defined direction, but also leave open the possibility of framing the algorithms in such a way that could be used as examples in the future.

One question I have is regarding the specifics of how to compare algorithms against each other. It seems like privacy loss would be a good measure, and I know how I would calculate it theoretically given a particular algorithm, but what would be a good way to measure performance of any given algorithm empirically?