Project Title: Local Model Heavy Hitters

Team Members: Yueran Cao & Kang Zhao

What are we planning to do?

We plan to implement and test a local-model algorithm for finding heavy hitters—the most frequent items in a dataset—under local differential privacy (LDP). In the local model, each user perturbs their data before sending it to the server, which means the server never sees the original values. This setup is different from the central model, where noise is added only after data collection.

Our goal is to simulate this process. Instead of creating an actual client-server system, we'll simulate both sides in one script. This idea came from our discussion with the instructor, and it makes sense since the "client" just performs randomization. Our simulation will create synthetic user data, add local noise (possibly using randomized response and hashing), and then aggregate the reports to try to recover the top-k frequent items.

We also plan to evaluate how well the algorithm works under different settings. We'll test:

- Different domain sizes (e.g., number of possible input values)
- Different user distributions (e.g., uniform vs. heavy-tailed)
- Different levels of privacy (epsilon values)

We've read the paper "*Practical Locally Private Heavy Hitters*" and some public code examples to guide our design. The goal is to understand how privacy affects the accuracy of detecting heavy hitters, and what kinds of trade-offs are involved.