4. **(12 pts.) Abstract Data Types**: .

Choose the best abstract data type (List, Set, Map, PQ, Queue, Stack) given the description of the desired data structures below. Be as specific as possible (don't answer List if a Queue is applicable). Also show what types would be used as template arguments (e.g. `map<string, int>` or `stack<double>`). If multiple "best" options exist, choose either. Give a **SHORT** 1-2 sentence justification for your choice (don't waste time on a long explanation...it won't help you get any more credit than a short answer).

4.1. An ADT to support the following: Given a city, retrieve the zip codes that are represented in the city's boundaries. *(Again, write your answers directly in Gradescope)*

Map<string, set<string or int> > (key = city, value = all zip codes in that city)
Wouldn't want a duplicate zip code so set is the best, but list can receive full credit

4.2. Normally customer service centers process customer calls in order of arrival. Research at USC allows voice analysis to understand the frustration level and emotion of a caller. Thus, a new system orders customers based on their frustration level. This requires changing the customer service software from using a _queue_ ADT (in the normal system) to a __priority queue___ ADT in the new system. No justification necessary (just fill in the blanks).

4.3. An ADT to store each of the individual items in a grocery purchase as they are scanned by the electronic reader.

List < string > (can replace "string" with "item" or something similar)
Also acceptable: Map<string, int> - String is item name, and int is quantity

4.4. An ADT to store the latest changes to **ALL** open documents in a word processor so that the user can, given the document name, undo the latest changes for each document. You may assume a "change" is just a string.

Map<string, stack<string> > (key=document name, value = stack of changes)
(partial credit List<changes> rather than Stack))