Time Limit: 60 Minutes

General Instructions

1. Attempt any **ONE** of the given problems
2. You **must** have a **working production quality code** by the end of this exercise
3. Feel free to use any tech stack of your choice

Problem Statements

1. Build a spell checker service, with following suggestions in mind:
    1. The service should expose an endpoint to perform the spell check operation. It should predict a set of corrected words sorted by their relevance – **This is the minimum your code should be able to do.**
    2. Can the dictionary (if required) be refreshed at runtime?
    3. If available, how would the service take into account the context info for more relevant predictions
    4. Can the service be internationalized?
    5. How well will the service scale?

2. Implement a multithreaded networked sorted set that handles positive integers. The server maintains multiple sorted sets and communicates via the protocol described below:
    1. ***Add Score*** Adds member `<key>` to `<set>`, with score `<score>`. If `<set>` doesn't exist, it's created. If `<key>` is already in `<set>`, its score is updated.
        a. Client: `<4> <1> <set> <key> <score>`
        b. Server: `<0>`
    2. ***Remove Key:*** Removes `<key>` from `<set>` if `<set>` exists and `<key>` is in `<set>`.
        a. Client: `<3> <2> <set> <key>`
        b. Server: `<0>`
    3. ***Get Size:*** Returns the size of set `<set>`, or 0 if `<set>` doesn't exist.
        a. Client: `<2> <3> <set>`
        b. Server: `<1> <size>`
    4. ***Get key-value:*** Returns the score of key `<key>` in `<set>`, and 0 if either the set does not exist or does not contain `<key>`.
        a. Client: `<3> <4> <set> <key>`
        b. Server: `<1> <score>`
    5. ***DISCONNECT****:*
        a. Client: `<1> <6>`
        b. Server: No response, then disconnect the client

6. ***[Bonus] Get Range:*** Returns all elements in sets <set1> ... <setM> with scores in the range [<lower>, <upper>]. Elements should be returned sorted by non-decreasing order of key. If two keys match, the elements with matching keys should be sorted by non-decreasing order of value.
    a. Client: <N> <5> <set1> ... <setM> <0> <lower> <upper>
    b. Server: <K> [<key> <score>] (repeat for each element of the set returned, where K is the total number of integers returned)

Protocol description

1. All parameters are **positive integers** separated by whitespace[s]
2. First integer describes the no of arguments to follow
3. Second integer represents the operation
4. Then follows the operation parameters
5. <set> is an integer id representing the set