RANGE QUERY FOR TEXT

There are many correct implementations of this problem; we are not looking for any single one of them, but just that your design is clean, efficient, and that the trade-offs you make are documented in the comments.

Your task is to implement, in C++, an API to track a set of text (ASCII characters only) ranges that supports as much of the functionality mentioned below as time permits.

PREMISE

A text range is defined as a set of strings using the familiar "bounds" notation. Bounds are marked by '[', ']', '(', and ')'. Square brackets denote an edge that is included in the text range, and parentheses denote an edge that is not included. Text ranges do not have to be continuous. For example, one such text range could be "[AaA - BaB]".

- A string "AA" would fall before this range.
- A string "Aaab" would fall inside this range.
- A string "Cblahblah" would fall after this range.

Addition of a range "[Aac - CaC]" to the above set would make the tracked set:

[AaA - CaC]

Addition of a range "[Dd - Df]" to the above set would make the tracked set:

[AaA - CaC], [Dd - Df]

Deletion of a range "[bb - CA]" from the above set would make the tracked set:

[AaA - bb), (CA - CaC], [Dd - Df]

EXERCISE

Provide APIs for:

1. Addition of a text range to the set being tracked.
2. Deletion of a text range from the set being tracked.
3. Query on whether a specific string is inside the set of ranges being tracked.

You do NOT need to make a program using this API very elaborate.

However, feel free to do so if it would be helpful to you in designing or debugging the API.

You may only use the C++ standard library and the pthreads library, but no others (Do not use STL). All source code must be your own.

Making the API thread-safe wherever appropriate would be a plus.

Please strive for:

1. Clean design — Make your code readable and re-usable where possible; add comments as appropriate; give thought to your interface,
2. Efficiency — Try to make each functionality as fast as possible,
3. Robustness — Handle error sensibly,
4. Correctness.

Please briefly explain your design choices and assumptions in your comments.