Darley Technologies - Round 1 Interview Questions

The solutions must be provided in Rust. Please mention all your steps and explain what led you to choose your solution. You can briefly comment on other solutions and ideas which you had while solving this task.

Part 1: Self-Evaluation

On a scale from 1 to 10, rate your experience with Rust and how idiomatic your Rust code typically is.

Part 2: Data Structures and Algorithms

Create a data-set of words from the book *https://www.gutenberg.org/files/98/98-0.txt*. Implement a fixed sized open addressing hash table by using linear probing to resolve collisions. Assume that the keys are the words from the given data-set and the hash table's values are integers. You need to implement the following functions with O(1)-complexity:

- `insert(key, value)`
  `inserts a new key-value pair or replaces a key's existing value,`
- `remove(key)`
  `removes the corresponding key-value pair,`
- `get(key)`
  `returns the value of the corresponding key,`
- `get_last()`
  `returns the most recent key-value pair that was either inserted or updated and is still present,`
- `get_first()`
  `returns the least recent key-value pair that was either inserted or updated and is still present`

Part 3: Trading Specific Algorithms

Review the Binance European Options API documentation at https://binance-docs.github.io/apidocs/voptions/en/.

Then:

- Retrieve data from the endpoint:
  `GET /eapi/v1/ticker`
- Write a parsing algorithm for the instrument statistics and print the result
- Measure the parsing speed for a single entry and evaluate the algorithmic complexity of your approach. Document any potential optimizations aimed at achieving low latency.