Naveen Dharshanala

16351902

Interaction with the LLM

At first, I asked the LLM for a Racket-based quicksort algorithm that used the "median of medians" method to select pivots. The primary emphasis of the code given by the LLM incorporated the utilization of underlying library capabilities, especially the sort capability, which was improper because of the task prerequisite to keep away from worked in arranging schedules. The LLM was alerted to the problem by me, and as a result, the code was improved to eliminate any library calls that handled sorting internally.

Following that, the LLM offered a solution that utilized sort in the median-finding section while still employing the median-of-medians strategy. Right now, I requested further explanation and mentioned the LLM to adhere to simply custom rationale. The objective was to completely avoid Racket's built-in sort for the core sorting mechanism, so this was a necessary step.

In response, the LLM attempted to develop a bespoke partitioning-based sorting logic that would divide lists into sublists with five elements, locate the median of each sublist, and then recursively locate the median of medians. While this approach was doing great, the execution had a couple of difficulties with edge cases and the size of sublists. I saw that for records less than five components, the apportioning rationale wasn't taking care of things accurately. I provided this feedback and requested that the LLM improve its partitioning and handling of lists smaller than five.

I ran tests to ensure that the code was correct after dealing with the edge cases and improving the way sublists were handled. The custom median-of-medians method was used by the code to sort lists correctly. As a stress test to ensure the algorithm's efficiency and robustness, the test cases included lists of varying sizes, from small lists with four elements to large lists with over a million elements.

Discussion of the Code

Refinement Process Initial Code Attempt The LLM used Racket's built-in sort function in the find-median function for the initial code attempt. This was a problem because the task required a completely unique sorting method. The algorithm's overall structure and concept—namely, partitioning and finding medians—were correct; however, it was necessary to eliminate the use of the built-in sorting function. This prompted the main huge refinement demand.

Adjustments to Partitioning

Subsequent to bringing up the issue with sort, the LLM changed the code by carrying out a segment into-sublists capability that would part the information list into sublists of size five or less components. Recursive partitioning was used by the LLM, but it initially failed to handle lists with fewer than five elements because it tried to take sublists of a fixed size regardless of the length of the list. During testing, I discovered this issue and asked the LLM to make sure that the partitioning logic handled lists of varying lengths, especially those with fewer than five elements, correctly. The LLM changed the function so that the size of the sublist can change on the fly.

Further Refinements to Middle Estimation

Following this, the find-middle capability was fixed to work absolutely on arrangements of up to five components by physically arranging them inside the capability. Nonetheless, after extra testing, I saw that the LLM battled with taking care of situations where records contained less components than anticipated, particularly while recursively ascertaining the middle of medians. I gave criticism to additionally refine the recursive design with the goal that more modest records were taken care of all the more smoothly.

Final Code Testing

All of the necessary test cases were passed by the final working version of the code. I ran the quicksort capability on a few arrangements of various sizes, incorporating a pressure test with a rundown of more than 1,000,000 components. The code performed satisfactorily and satisfied the sorting requirements without utilizing any built-in sorting functions. Using a custom median-of-medians pivot selection method, the final quicksort implementation was able to correctly sort lists.