Prompt Engineering Frameworks

Roy Van

Part 1:

Task: I want AI to explain the time complexity of C++ codes using Big O notation. Analyze its complexity and efficiency variations with different inputs and provide an optimized sorting algorithm.

Context: We are analyzing a sorting algorithm implemented using Selection Sort. Selection Sort takes the minimum value in an array repeatedly and switches its position with the first unsorted term. This will continue until the entire array is sorted.

References: Here is an example of a C++ code of selection sort:

#include <iostream>

using namespace std;

void selectionSort(int arr[], int n){

for(int i = 0; i < n; i++){

for(int j = i + 1; j < n; j++){

if(arr[i] > arr[j]){

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

// swap(arr[i], arr[j]);

}

}

}

}

int main(){

int arr[] = {5, 5, 4, 2, 5, 6, 3, 4};

int n = sizeof(arr) / sizeof(arr[0]); // # of elements in the array

selectionSort(arr, n);

// Print the resulting array

for(int i = 0; i < n; i++){

cout << arr[i] << " ";

}

cout << endl;

return 0;

}

Evaluate: I will evaluate AI’s response based on the aspects below:

* Does AI identify the time complexity of selection sort correctly? The answer should be O(n²) due to the two loops within the codes. Does AI also explain the reason of Big O of selection sort being O(n²) and make connections to the example I have provided?
* Does AI analyze the best, average and worst case of selection sort? Their Big O should all be the same which is O(n²).
* Suggestions for further improvements. To optimize the efficiency of sorting algorithms, what are some optimizations I can do? Does AI suggest better sorting algorithms such as quick sort and merge sort which has an average complexity of O(n log n) and explain why they are better algorithms for larger scale inputs?

Iterate: If AI’s outputs are not ideal, I will make the following refinements.

* I could ask for a more detailed report about in the best-case scenario, although fewer swaps are made, how selection sorts behave and affects their overall time complexity.
* I could specify that the AI should make comparisons between more advanced sorting algorithms such as merge sort and quick sort, and ask for their performances under extreme conditions.
* As an additional layer, I could ask for information on analyzing selection sort’s space complexity(which is known to be 1).

Paet 2:

文本

AI 生成的内容可能不正确。 文本

AI 生成的内容可能不正确。

AI Response:

文本

AI 生成的内容可能不正确。 文本

AI 生成的内容可能不正确。 文本

AI 生成的内容可能不正确。 文本

AI 生成的内容可能不正确。

Chosen LLM: ChatGPT 4o

电脑屏幕截图

AI 生成的内容可能不正确。

From this leaderboard (<https://lmarena.ai/?leaderboard>), I can see that ChatGPT 4o by OpenAI is very good at answering coding related questions, this is the reason I pick this model to assist my computer science learning.

References:

<https://openai.com/index/gpt-4-research/>

<https://arxiv.org/abs/2303.12712>

<https://lmarena.ai/?leaderboard>

With respect to its accuracy and completeness, with some refinements based on original input, ChatGPT 4o is able to provide detailed and well-organized answers that matches my prompt.

Part 3 Reflection:

I chose my special structure for prompts to ensure that artificial intelligence can provide a versatile, well-considered response to my questions. I ask AI to conduct a thorough analysis on ‘Selection Sort’, this includes calculating and comparing the time complexity and space complexity under best case, average case and worst-case scenarios. I also asked AI to make comparisons between other efficient and complicated sorting algorithms such as merge sort and quick sort, as well as comparing their complexities. I ensured that the response covers as much useful knowledge as possible by refining my prompts through adding in more specific requirements and possibilities, so that ChatGPT 4o can better understand my prompts and provide more practical and innovative information.

Five steps framework is highly helpful for me to explore the creation of a focused prompt. It helped clarify my communication when working with AI, making sure that I am getting a thoughtful response instead of any irreverent and superficial answers. Through this exercise, I realized that with cooperating with AI, clear and logical prompts are crucial. Since when we are dealing with coding issues, or analyzing algorithms, understanding how to craft effective prompts will also help in using AI as a tool for problem-solving and learning.