

Algorithm deconstruction challenge

Task priority sorting and filtering algorithm

Introduction

For this exercise, i analyzed the task priority sorting and filtering algorithm. The purpose of this algorithm is to calculate an importance score for every task and sort task so the most important ones appear first. Instead of only using priority level, it considers several factors, such as dues dates, task status, tags, and recent updates. This creates a smarter way of organizing tasks.

How the Algorithm Works

The algorithm follows these steps:

1. Assign a base score according to the task priority.
2. Increase the score if the task is due soon or already overdue.
3. Reduce the score if the task has already been completed or is currently under review.
4. Increase the score if the task contains important tags such as critical, urgent, or blocker.
5. Increase the score slightly if the task was recently updated.
6. Calculate the final score.
7. Sort all tasks from the highest score to the lowest score.

Start

|



Read Task Information

|



Assign Base Priority Score

|



Check Due Date

(Add bonus if overdue or due soon)



Check Task Status

(Reduce score if completed or under review)



Check Tags

(Add score for urgent or critical tags)



Check Recent Updates

(Add small bonus)



Calculate Final Score



Sort Tasks (Highest Score First)



Display Sorted Tasks

Reflection Questions

1. How did the AI's explanation change your understanding of the algorithm?

The AI made the algorithm much easier to understand by explaining it step by step instead of only showing the code. At first, I only saw different calculations and conditions, but I did not understand why they were there. After the explanation, I realized that each condition contributes to calculating a final importance score that determines the order of the tasks. The AI also explained how different factors work together to prioritize tasks, making the logic much clearer.

2. What aspects were still difficult to understand after the AI explanation?

Although the AI explained the algorithm well, I still found it slightly difficult to understand how the scoring values were chosen. For example, I was unsure why overdue tasks receive a bonus of 35 points while recently updated tasks receive only 5 points. I also wanted a deeper explanation of how these values would be selected in a real software project. However, I understood the overall purpose of the algorithm.



3. How would you explain this algorithm to another junior developer?

I would explain that the algorithm works like a ranking system. Every task starts with a score based on its priority level. The score then changes depending on other important information, such as how close the deadline is, whether the task has been completed, whether it has important tags, and whether it has been updated recently. After all the scores are calculated, the tasks are sorted so that the highest-scoring task appears first. This helps users focus on the most important work before less urgent tasks.

Insights and Learning Points

During this exercise, I learned that algorithms are not only about solving problems but also about making better decisions using different conditions. I also learned that one task can have multiple factors affecting its importance, rather than relying on a single priority value. AI helped simplify the code into understandable language, making it much easier to follow the logic behind the algorithm. I also realized that breaking code into smaller sections is one of the best ways to understand unfamiliar programs.

Conclusion

This exercise improved my understanding of how task prioritization algorithms work in real applications. I now understand that software can use multiple rules to calculate the importance of tasks instead of relying on simple priority levels. Using AI alongside the source code helped me understand the algorithm faster and increased my confidence in reading and explaining unfamiliar code. I will continue using this approach when learning new algorithms and software projects in the future.