

The AI-generated version is more concise and efficient. It uses Python's built-in `sorted()` function with a lambda expression, making the code both readable and optimized. It leverages Python's internal Timsort algorithm ($O(n \log n)$) and requires only one line, which reduces the likelihood of human error.

In contrast, the manual version uses a custom insertion-based sorting algorithm. While functionally correct, it has a worst-case time complexity of $O(n^2)$, which is inefficient for large datasets. The manual approach is also harder to read and maintain, with more opportunities for bugs due to its complexity.

The AI-suggested code clearly shows how code generation tools like GitHub Copilot or Tabnine can speed up development by reducing boilerplate and leveraging best practices. However, understanding the manual approach is still valuable, especially when explaining or debugging complex behaviors.

Conclusion:

The AI-suggested code is more efficient, readable, and production-ready, making it preferable for real-world development — though developers should always review and understand AI outputs before using them blindly.