

Palindrome Product Time-Complexity Analysis

Jervani Z Thompson

October 6, 2023

Introduction

This document analyzes a method to find the largest palindrome that is a product of two 3-digit numbers. A palindrome is a number that reads the same forwards and backwards. The original problem can be stated as follows:

"Find the largest palindrome made from the product of two 3-digit numbers."

Method

The method involves iterating over 3-digit numbers in reverse order, calculating their product, and checking if the product is a palindrome. The steps are outlined below:

1. Initialize a variable `max_palindrome` to zero. This variable will hold the largest palindrome found.
2. Iterate over 3-digit numbers in reverse (from 999 to 100) using two nested loops. The inner loop should iterate from the current value of the outer loop variable down to 100.
3. During each iteration, calculate the product of the two numbers. If the product is a palindrome and larger than `max_palindrome`, update `max_palindrome`.
4. (Optimization) In the inner loop, if the product of the outer loop variable and the inner loop variable is smaller than `max_palindrome`, break out of the inner loop.
5. After the loops finish executing, `max_palindrome` will contain the largest palindrome made from the product of two 3-digit numbers.

Time Complexity Analysis

The time complexity of the method is primarily determined by the two nested loops that iterate over 3-digit numbers:

- The outer loop iterates over all 3-digit numbers, leading to a time complexity of $O(900)$, as it iterates from 100 to 999 inclusive.
- The inner loop also iterates over 3-digit numbers, with an optimization that breaks early if the product is smaller than the current maximum palindrome. In the worst case (without considering the optimization), the inner loop would also have a time complexity of $O(900)$.
- Since the inner loop is nested within the outer loop, their time complexities are multiplied, leading to a worst-case time complexity of $O(900 \times 900)$ or $O(810,000)$, which is simplified to $O(n^2)$, where n is the number of 3-digit numbers.
- Due to the optimization (breaking the inner loop early), the actual time complexity is less than the worst-case scenario.

Conclusion

The implemented method efficiently finds the largest palindrome made from the product of two 3-digit numbers. The time complexity of the method is $O(n^2)$ in the worst case but is often better in practical applications due to the optimization implemented in the inner loop.