

Algorithm Analysis

Abraham J. Reines

April 12, 2024

Task 6.3.1

In the examination of the asymptotic behavior of the function $f(n) = 10^{12}n^3 + 10^6n^2 + n + 1$ against the polynomial $g(n) = n^4$, we have applied theoretical analysis complemented by limit calculations using SymPy to derive the following conclusions:

1. **Is $10^{12}n^3 + 10^6n^2 + n + 1$ not equivalent to $\Theta(n^4)$?**
Yes. $10^{12}n^3 + 10^6n^2 + n + 1$ is not $\Theta(n^4)$ because the highest order term, n^3 , dictates the growth rate of the function. As n approaches infinity, the n^4 term would grow faster than the n^3 term. Hence, $10^{12}n^3 + 10^6n^2 + n + 1$ is $\Theta(n^3)$, not $\Theta(n^4)$.
2. **Is $10^{12}n^3 + 10^6n^2 + n + 1$ equivalent to $o(n^4)$?**
Yes. $10^{12}n^3 + 10^6n^2 + n + 1$ is $o(n^4)$ because as n grows, the function $10^{12}n^3 + 10^6n^2 + n + 1$ increases at a slower rate than any function proportional to n^4 . Thus, it can be bounded above by $C \cdot n^4$ for some constant C and sufficiently large n .
3. **Is $10^{12}n^3 + 10^6n^2 + n + 1$ equivalent to $\omega(n^4)$?**
No. $10^{12}n^3 + 10^6n^2 + n + 1$ is not $\omega(n^4)$ because it does not grow faster than any function proportional to n^4 . For $f(n)$ to be $\omega(g(n))$, $f(n)$ must eventually exceed $C \cdot g(n)$ for any constant C , which is not the case here as n^3 is the dominant term in $f(n)$, and it grows slower than n^4 .

These findings underscore the importance of both theoretical and computational methods in the analysis of algorithmic complexity.

6.3.2

Application Case:

- 250GB hard disk with 160GB of files.
- CPU with one core, 4GHz clock speed, and 64-bit registers.
- Capable of 4×10^9 8-character comparisons per second.

Search Time Analysis

1. Single 1KB Signature:

The time required to search with a single 1KB signature is computed to be approximately 5 seconds. This duration represents the worst-case scenario, providing a concrete timeframe for completing a single search cycle. It is important to note that the Boyer-Moore algorithm will often perform faster than this worst-case estimate due to its pattern-skipping feature.

2. Database of 1 Million 1KB Signatures:

Scaling up the search to account for a database of 1 million 1KB signatures, the total time estimated for the search is approximately 57.87 days, assuming a worst-case scenario where each signature search is independent and linear. The actual performance of the Boyer-Moore algorithm is expected to be better due to its sublinear average-case performance.

3. Improving Efficiency:

To further improve the efficiency of malware/virus detection through string matching, we can utilize parallel processing by distributing the search workload across multiple CPU cores or even different machines. Additionally, implementing more advanced string matching algorithms such as Aho-Corasick, which is designed for searching multiple patterns simultaneously, can significantly reduce the time complexity. Optimizing the algorithm to take advantage of the preprocessing of the virus signatures and employing heuristic methods to skip unlikely sections of data can also enhance performance.

Scripting

References

Academic Integrity Pledge

“This work complies with the JMU honor code. I did not give or receive unauthorized help on this assignment.”