Miguel Sort: A Revolutionary $O(n + 2^{32})$ Sorting Algorithm for Maximum Inefficiency

Pranav Jayakumar

February 5, 2025

Abstract

In this groundbreaking study, we present **Miguel Sort**, an innovative sorting algorithm that leverages a preallocated boolean array of 2^{32} elements to efficiently sort five integers in just 15 minutes and 80GB of RAM. This paradigm-shifting approach challenges conventional wisdom by demonstrating that *Big-O notation is merely a suggestion.* We explore Miguel Sort's applications in memory exhaustion, computational inefficiency, and causing unexpected system crashes in cloud environments.

1 Introduction

Sorting algorithms have long been a cornerstone of computer science, with well-established methods such as QuickSort, MergeSort, and the notoriously sluggish Bubble Sort. However, none of these algorithms truly embrace the full potential of computational inefficiency. This paper introduces Miguel Sort, an algorithm designed with one goal in mind: **to push the limits of impracticality**.

2 Algorithm Description

Miguel Sort operates using the following steps:

1. Allocate a boolean array of size 2^{32}, requiring approximately 80GB of RAM.
2. Iterate through the input array, marking indices in the boolean array as `True`.
3. Iterate through all 2^{32} indices and reconstruct the sorted array.
4. Crash the system if memory runs out (this step is unavoidable).

3 Time and Space Complexity

Miguel Sort achieves an impressive complexity of:

- **Time Complexity**: $O(n + 2^{32})$, ensuring a runtime that scales exponentially worse than any practical sorting method.
- **Space Complexity**: $O(2^{32})$, making it ideal for exhausting memory resources.

4 Experimental Results

Empirical tests were conducted on a high-performance Google Colab instance. The results were as follows:

- Sorting a set of **5** integers: **15 minutes, 80GB RAM usage**
- Sorting 10 integers: **Colab session forcibly disconnected**
- Sorting negative numbers: **Immediate segmentation fault**

5 Conclusion

Miguel Sort sets a new standard in computational inefficiency. While traditional algorithms focus on speed and resource optimization, Miguel Sort boldly demonstrates that these concerns are *entirely optional*. We believe this work opens new research avenues in the field of algorithmic absurdity and may inspire future developments in memory-intensive sorting techniques. Future work includes GPU acceleration and a blockchain implementation for maximum impracticality.

Acknowledgments

The authors wish to thank the computer science community for their patience as we push the boundaries of what should never be done.