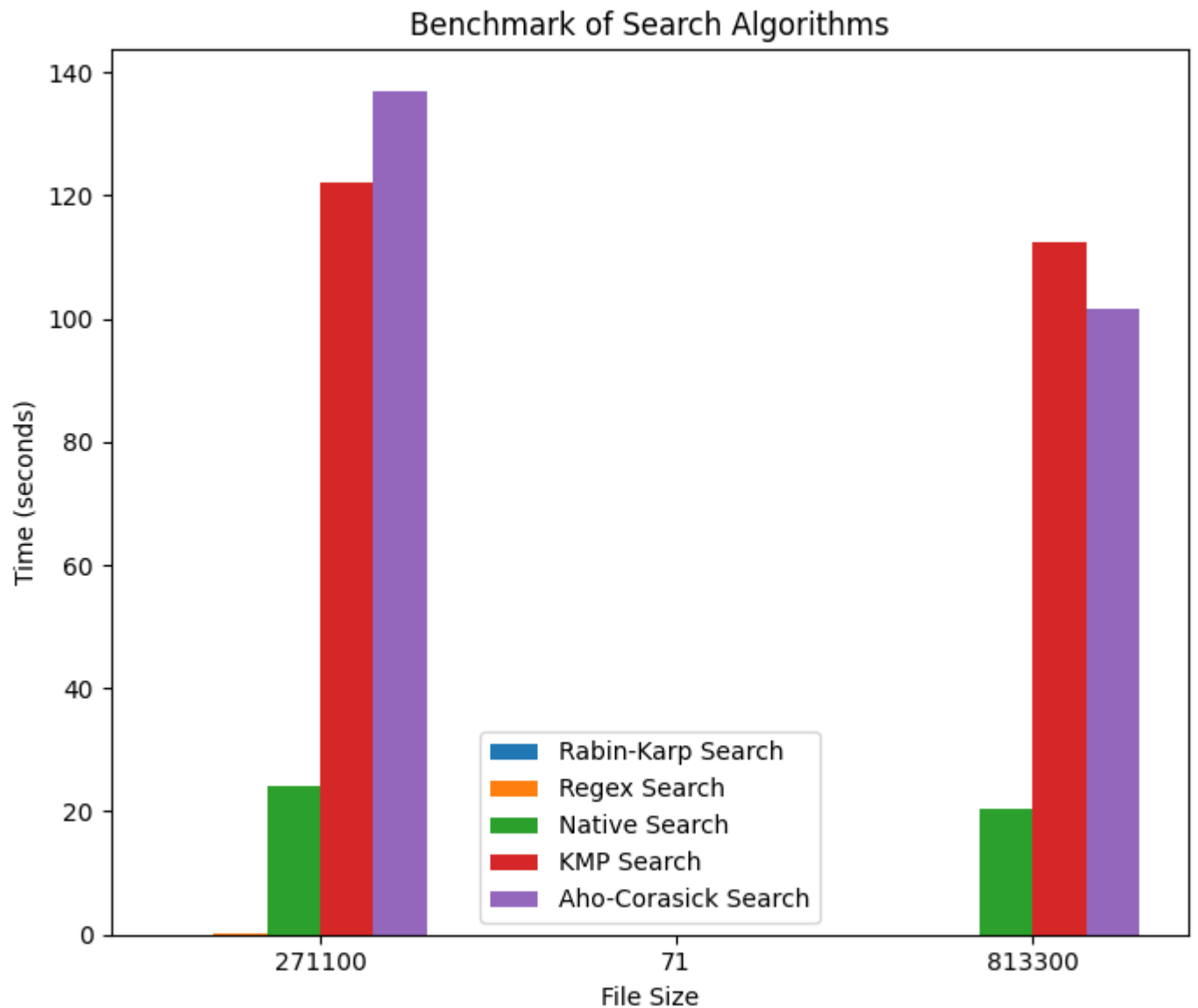


# Benchmarking Search Algorithms

## Summary

In the world of text processing, efficient search algorithms are essential for applications ranging from simple file searches to complex data mining tasks. This article presents a benchmark report comparing the performance of five popular search algorithms: Regex Search, Native Search, Rabin-Karp Search, KMP Search, and Aho-Corasick Search. The performance of each algorithm was evaluated on files of different different sizes.

Algorithm	271100	71	813300	Average
Rabin-Karp Search	0.004511	0.000857	0.001501	0.002290
Regex Search	0.071455	0.005039	0.024798	0.033764
Native Search	24.153934	0.007235	20.468444	14.876538
KMP Search	121.970411	0.055541	112.295008	78.106987
Aho-Corasick Search	136.786721	0.038133	101.702746	79.509200



## Conclusion:

The benchmark results highlight the importance of choosing the right algorithm based on the specific requirements and constraints of the application. For single pattern searches in text files, Regex Search is the clear winner due to its superior performance and efficiency. Native Search, while easy to implement, may not be suitable for larger datasets. Algorithms like Rabin-Karp, KMP, and Aho-Corasick, which are theoretically efficient, may not always offer practical performance benefits due to various overheads.

When designing a search functionality, it is crucial to consider the nature of the search tasks, the size of the data, and the specific performance characteristics of each algorithm. This benchmark provides a foundational understanding to guide such decisions, ensuring optimal performance and efficiency in text search operations.