Reach new heights of peak performance with this new word processor feature!!! Remember those old days of searching through your twenty page plus documents in pursuit of a simple fact, citation, or just to simply recall what you wrote on page five, paragraph three, in the second sentence? Or perhaps you’re still doing that! If you switch over to our word processor, you are promised quicker access to the information that you need! In fact, the larger, more detailed the document you create, the more it will outlast and outperform the slow, grinding, atrocious word processor you are using right now!!! No joke, through some of our detailed and thought-out tests, our research team was able to reach speeds 20x faster than the competitors! Now who would not want that? Who is sitting at work, home, or school content with a slower piece of software that is a clear detriment to the productivity and progress of completing a document for one’s boss, a paper for a professor, or even just a simple diary or journal? And the best part? It comes at no additional cost to you! Yep, the team behind the development for this fantastic feature kept you, the consumer, in mind. Surely that keeps your mind at rest knowing that you will not be ripped off, your money can stay in your wallet, and your life can only progress to a higher state of productivity due to this word processor’s innovation!! In addition, this feature will prevail through the test of time. Regardless, several years down the road, future word processors will possess the same innerworkings that lead to this huge performance boost, but by that time they will be charging a premium, tricking you into believing it is new technology!! We both know you must search your document for words or phrases at some point and you do not want that to be the reason why your word processor is so slow. With our innovative algorithm, you will never have to worry about that affecting your opinion of your word processor; it is practically unnoticeable! You have nothing to lose and everything to gain, so go ahead, buy our word processor with this cutting-edge searching algorithm. We guarantee you will never regret this decision, and your family, friends, and colleagues will be jealous that you made such a smart choice!!!!!

**Justifications**

The first claim in the advertisement is “the larger, more detailed the document you create, the more it will outlast and outperform…”. This is based on the fact that the runtime for Boyer-Moore in the best case is O(n/m), whereas KMP is O(n+m) and Rabin-Karp is O(n+m), where n is the length of the document in characters and m is the length of the pattern. So, because of this, following from simple algebra, as n increases, the runtime of Boyer-Moore will prove to be more efficient compared to the other pattern matching algorithms.

Second, “…our research team was able to reach speeds 20x faster than the competitors”. Because this is 1977, and other competing string pattern matching algorithms are currently being used, such as KMP, Boyer-Moore must show it is faster. For a document where m = 20, in the best case, as mentioned above, Boyer-Moore will be twenty times faster regardless of the length of the document. An m of twenty is a realistic value that shows its superiority over competitors, where a value of one hundred is too large and a value of two is rather tiny.

Next, the claim of “It comes at no additional cost to you!” is made. This is a clear and easily justifiable statement. The Boyer-Moore string pattern matching algorithm is a production of research released to the academic world free of charge. Because of this, anybody may implement the algorithm in their software, ridding of any additional cost in terms of physical money.

The fourth claim is “…this feature will prevail through the test of time”. Not only is the best-case runtime of Boyer-Moore better than KMP and Rabin-Karp, but the average case is on par with both while KMP is the only one with O(n) worst case, since the other two algorithms are O(nm). The worst case will rarely occur because it requires a shift of one to the right for every iteration, therefore on average Boyer-Moore will perform equal to or better than its competitors. Lastly, Boyer-Moore may theoretically be the same runtime as KMP in the average case, but in practice its performance is better, similar to comparing quicksort and mergesort. As a result, this algorithm will end up being used in all kinds of software until a newly researched pattern matching algorithm is researched, which even then will have a difficult time beating Boyer-Moore.

Finally, the claim “…future word processors will possess the same innerworkings …” is advertised. As discussed several times already, Boyer-Moore outperforms other algorithms. Its popularity will inevitably surpass all string pattern matching algorithms and most, if not all, programs requiring pattern matching will have it implemented within their code.

Overall, these claims dealt with runtime and efficiency of Boyer-Moore over its competitors. It is clear from its performance, zero cost, and eventual rise of popularity that this algorithm will be widely used across most software in the world to deal with string pattern matching. To add onto everything mentioned, Boyer-Moore’s implementation is trivial; it does not take an expert of computer science to understand and write the code, so that will not be a problem for developers. This algorithm clearly shows its usability and power and is bringing a faster, more efficient future of string pattern matching to the world.