

Joint Comparative Analysis Report.

Pair 3.

Student A: Berdaly Nurila (Boyer-Moore Majority Vote)

Student B: Ismagambetova Fariza (Kadane's algorithm)

This report compares two linear array algorithms:

- 1) **Boyer-Moore Majority Vote** finds an element that occurs more than $n/2$ times.
- 2) **Kadane's Algorithm** finds the subarray with the maximum sum.

How each algorithm works:

- 1) Boyer-Moore algorithm maintains a candidate element and a counter while scanning array once. When current element matches the candidate, the counter increments, otherwise it decrements. And if counter becomes zero, the current becomes the new candidate.
- 2) Kadane's algorithm maintains currentMax value which is the best subarray sum ending at current position; and globalMax – the best subarray sum found overall. For each element algorithm decides whether to extend the current subarray or start new one. It runs a single pass without recursion, and outputs maximum sum and its start and end indices.

Complexity Comparison.

	Boyer-Moore algorithm	Kadane's algorithm
Best-case Time	$\Omega(n)$	$\Omega(n)$
Average-case time	$\Theta(n)$	$\Theta(n)$
Worst-case Time	$O(n)$	$O(n)$

Both algorithms have optimal linear time with constant space complexity.

Kadane's algorithm perform more arithmetic per iteration, while Boyer-Moore executes simpler operations, giving it a bit smaller constant factors in practice.

Empirical validation.

Empirical measurements were obtained for $n = 100, 1\,000, 10\,000, 100\,000$ elements.

	Array size	Execution time	Comparisons	Array Access
Boyer-Moore	100	168875	99	99
	1 000	27458	999	999
	10 000	568500	9999	9999
	100 000	2124542	99999	99999
Kadane's	100	10660000	198	198
	1 000	47100	1998	1998
	10 000	778700	19998	19998
	100 000	2838600	199998	199998

It shows that both algorithms grow linearly with n , confirms $\Theta(n)$ behavior.

Boyer-Moore executes roughly half as many operations and completes faster due to fewer arithmetic steps per iteration.

There are no non-linear big jumps, maintaining low overhead.

In general, the comparison:

- 1) Boyer-Moore detect majority element; Kadane's finds subarray with max sum.
- 2) Constant factor of Boyer-Moore is smaller due to simpler loop, while Kadane's is slightly larger due to two updates per i .
- 3) In terms of practical speed, Boyer-Moore is faster, Kadane's is slightly slower, but they still have similar trend.
- 4) They **both** go through array only once, and both have linear time and constant space.