

Cross-Review Summary

1. Name of algorithms

Kadane Algorithm	Nurtilek Kobylandy	Finds the subarray with the maximum sum
Boyer Moore Majority Vote Algorithm	Mansur Serikbai	Finds an element that occurs more than half of the time in an array

2. Complexity of algorithms

Algorithm	Time	Space	Comment
Kadane Algorithm	$O(n)$	$O(1)$	One pass through the array, only a few variables
Boyer Moore	$O(n)$	$O(1)$	Two passes through the array to check the candidate, optimized version - potentially early exit

3. Metrics and Performance

Kadane Algorithm - uses *metrics.comparisons* and *metrics.subarrayStartUpdates*. Tracks each element and updates the current sum. Logs and process well.

Boyer Moore - uses *metrics.comparisons* and *metrics.candidateUpdates*. The optimized version allows you to finish checking earlier if the majority threshold is reached. A fairly economical algorithm, without unnecessary variables or arrays.

4. Handling edge cases

Kadane Algorithm

- empty array → returns [0,-1,-1]
- all negative element → returns the maximum element
- supports arrays with zeros

Boyer Moore

- array is empty → returns -1
- no model element → returns -1
- the optimized version checks for early exit if the candidate cannot be selected

5. Potential optimization

Kadane Algorithm

- You can remove part of the print statement in the loop to speed things up on large arrays
- The *metrics.comparisons* metric can be extended to estimate the actual number of operations

Boyer Moore

- An optimized version with early release has already been implemented
- it is also possible to test large arrays in parallel for speed measurements

6. Applicability

Kadane Algorithm	Finance, Signal processing, Maximum substring problems
Boyer Moore	Voting, Statistical analysis, Search for model elements

7. Comparative table

Criteria	Kadane Algorithm	Boyer-Moore Algorithm
Aim	Maximum subarray sum	Element modul ($>n/2$)
Time complexity	$O(n)$	$O(n)$
Space complexity	$O(1)$	$O(1)$
Ease of optimization	Average	High that means already optimized
Code readability	Good with logs	Clean code easy to read
Handling edge cases	Complete	Complete
Metrics	comparisons, subarrayStartUpdates	comparisons, candidateUpdates

8. Conclusion

- 1. The Kadane Algorithm is suitable for finding the maximum substring in numerical arrays, is easy to track through logs and metrics, but for large arrays, printing must be disabled
- 2. Boyer Moore effectively finds the pattern element and is already optimized for early exit, making it more suitable for big data
- 3. Both algorithms have $O(n)$ time complexity and minimal memory consumption