

Astana IT University
Design and analysis of algorithm
Assignment 2

Group: SE-2401

Student: Tyulebayeva Arailym, Sabulla Diana

Algorithms: Kadane's Algorithm, Boyer-Moore Majority Vote

Problem Definition and Nature

Although both algorithms operate in linear time and are optimized for array processing, they address fundamentally different computational problems:

- **Boyer–Moore Majority Vote** solves a **decision problem**: it determines whether there exists an element that appears more than $\lfloor n/2 \rfloor$ times in the input array. The result is binary — either a majority element exists, or it does not.
- **Kadane’s Algorithm**, in contrast, solves an **optimization problem**: it finds the contiguous subarray with the maximum possible sum. The result is not merely a decision but a concrete subarray defined by its starting and ending indices and the sum of its elements.

Theoretical Complexity Comparison

Aspect	Boyer-Moore Majority Vote	Kadane's Algorithm
Time Complexity	$O(n)$	$O(n)$
Space Complexity	$O(1)$	$O(1)$
Passes Over Array	2(one for finding, second for verifying)	1
Best Case	$O(n)$	$O(n)$
Worst Case	$O(n)$	$O(n)$

Both algorithms achieve optimal asymptotic performance, running in linear time with constant space. While Boyer–Moore require a second pass to verify the majority candidate, this does not change its overall complexity class. The differences in iteration patterns are due to their distinct objectives rather than inefficiencies.

Strengths and Practical Advantages

Boyer–Moore Majority Vote is a very simple and efficient algorithm. It uses only two variables (candidate and count) and can find the majority element in just one pass through the array. This makes it great for:

- Streaming data processing
- Real-time systems
- Situations where memory usage must be very low

Kadane's Algorithm gives more detailed results by returning both the indices and the sum of the maximum subarray. It also:

- Works well with negative numbers
- Can be adapted for circular or multidimensional arrays
- Is useful for analyzing maximum segments in data

Limitations and Trade-offs

Boyer–Moore:

- Requires a **verification pass** when the existence of a majority element is not guaranteed.
- Works only when the frequency threshold is strictly greater than $n/2$ — it cannot detect **plurality** elements or minority patterns.

Kadane:

- May suffer from **integer overflow** when dealing with extremely large sums (mitigated by using long).
- Restricted to **contiguous subarrays** and cannot detect non-contiguous patterns.

Practical Applications and Use Cases

In real-world scenarios, the two algorithms serve very different purposes:

- **Boyer–Moore** is widely used in **streaming analytics**, **online processing**, and **voting-based systems** — for example, in leader election protocols, anomaly detection, or majority consensus mechanisms.
- **Kadane's Algorithm** is a foundational tool in **signal processing**, **financial analytics**, and **machine learning**, where maximum-sum subarrays represent optimal windows (e.g., periods of peak profit or high activity).

Comparative Conclusion and Insights

Despite their shared linear time complexity and constant space usage, the objectives and outcomes of these algorithms differ significantly. **Boyer–Moore** excels in decision-making tasks with minimal overhead, where detecting the presence of a majority is crucial. **Kadane's Algorithm**, on the other hand, provides deeper analytical insights into data structure and distribution by identifying optimal subarrays. Together, they demonstrate the diversity and versatility of linear-time algorithmic solutions in modern computational design and practice.