Pair submission(may be report)

Daniil Khvan

Aldiyar Zhaksylyk

1. Overview of Algorithms

Kadane's Algorithm (my work):

This algorithm finds the maximum sum of a subarray in an array. It also keeps track of where the subarray starts and ends. It works in one pass, updating a running sum and checking if it's bigger than the max. I also handled cases like when all numbers are negative.

Boyer-Moore Majority Vote (partner's work):

This one finds the majority element (the one that appears more than half the time). It keeps a candidate and a counter. If the counter drops to zero, it changes the candidate. To be safe, we need a second pass to check if the candidate is really the majority.

2. Complexity

- Time:
  - Kadane's: $\Theta(n)$, because we scan once. About 2 comparisons per element.
  - Boyer-Moore: $\Theta(n)$, also one pass. With verification, it's two passes but still linear.
- Space: Both use only a few variables ($\Theta(1)$). Kadane's stores indices and result array (a tiny constant). Boyer-Moore only uses candidate and count.

So both are linear time and very memory efficient.

3. Implementation and Optimizations

Kadane's (reviewed by partner):

- Code is clean, has comments and tests for edge cases (empty, null, negatives).
- Also has CLI and metrics tracking (comparisons, accesses).

- Improvements suggested: check for overflow (maybe use long).

Boyer-Moore (reviewed by me):

- Original code was simple but missing tests and input validation.
- I added verification step and metrics tracking.
- Suggested adding tests (like when no majority exists).

4. Empirical Results

- Both were tested with arrays of size 100 → 100,000.
- Kadane's: Time grew linearly (e.g., ~0.5ms at n=1000, ~53ms at n=100,000).
- Boyer-Moore: Also linear, but a bit slower with verification (~1ms at n=1000, ~96ms at n=100,000).
- Plots show straight lines (linear growth).

So both match theory: Θ(n). Kadane's a little faster since it doesn't always need a second pass.

5. Comparison

- Similarities: Both are linear time, constant space, and good for large data.
- Differences:
    - Kadane's solves "best sum" problem.
    - Boyer-Moore solves "most frequent element" problem.
    - Kadane's always works in one pass. Boyer-Moore sometimes needs a second pass.

6. Conclusion

Both algorithms are efficient and simple. Kadane's is already near-optimal and well-tested. Boyer-Moore needed improvements (verification, tests), but now it's also robust.

In practice: Kadane's is a bit faster, but Boyer-Moore solves a different problem and still works in Θ(n) with very low memory.