The Peak Profitability Problem at Chronos Capital

Story

At Chronos Capital, a leading financial analytics firm, data scientists analyze historical market data to identify patterns. They are currently working with a sequence representing the daily **profit or loss** of a major stock index over a long period. A positive value indicates a profitable day, while a negative value signifies a loss. The firm's goal is to find the single most profitable **continuous trading period** in the history of this index. This means identifying a "buy" date and a subsequent "sell" date such that the cumulative profit during that timeframe is maximized. A brute-force check of every possible start and end date is computationally expensive and too slow for their real-time analytics platform, which needs to process vast amounts of data efficiently. They require a more sophisticated algorithm to find this peak profitability window.

Task (Problem Definition)

You are given a sequence of N integers, where each integer represents the profit or loss on a given day. Your task is to find the **contiguous subarray** within this sequence that has the largest sum and return that sum.

- **Solution Requirement:** The time complexity of your solution must strictly be **O(n log n)**. Simple nested-loop solutions with $O(n^2)$ complexity will not be accepted as they will time out under the given constraints.
- **Required Output:**
  - An integer representing the sum of the most profitable contiguous subarray.

Example Scene

Scene 1 Daily Returns Sequence: [-2, 1, -3, 4, -1, 2, 1, -5, 4] Expected Maximum Profit: `6`. *Explanation (The Most Profitable Period):*

- By analyzing all possible continuous periods (subarrays), we find that the period starting from the 4th day and ending on the 7th day yields the highest profit.
- The subarray corresponding to this period is **[4, -1, 2, 1]**.
- The cumulative profit calculation for this period is:
  - $4 + (-1) + 2 + 1 = 6$
- No other contiguous subarray in the sequence has a sum greater than 6. For instance, the entire array sums to 1. The subarray [1, -3, 4] sums to 2.

Final Maximum Profit: 6 .