Kadane’s Algorithm

solved by	Senan
Platform	GeeksForGeeks
difficulty	Medium
tags	Kadane
language	C++
solved on	@24/11/2024
link	https://www.geeksforgeeks.org/problems/kadanes-algorithm-1587115620/1
Completion	☑

Intuition

The goal is to find the maximum sum of any contiguous subarray. This problem can be efficiently solved using **Kadane's Algorithm**, which iteratively computes the maximum sum ending at each position and keeps track of the overall maximum sum. The intuition is that at any index, we can either start a new subarray or extend the current subarray.

Approach

1. Start with the first element of the array as both the current sum (`sum`) and the maximum sum (`maxSum`).
2. Iterate through the array starting from the second element.
3. For each element, decide whether to include it in the existing subarray or start a new subarray. This decision is made by taking the maximum of the current element and `sum + arr[i]`.
4. Update the overall maximum sum (`maxSum`) by comparing it with the current sum.
5. Return `maxSum` after processing all elements.

Complexity

Time Complexity:

O(n) – The array is traversed once, and each element is processed in constant time.

Space Complexity:

O(1) – No additional space is used apart from a few variables.

Code

```
class Solution {
public:
    int maxSubarraySum(vector<int> &arr) {
        int sum = arr[0];
        int maxSum = sum;

        for(int i = 1; i < arr.size(); i++) {
            sum = max(arr[i], sum + arr[i]);
```

```
            maxSum = max(maxSum, sum);
        }

        return maxSum;
    }
};
```