DSA Study Notes Day 10:

Kadane's Algorithm (Maximum Subarray Problem)

Introduction to Kadane's Algorithm

- Kadane's Algorithm is a popular method to solve the **Maximum Subarray Problem**.
- It aims to find the largest sum of a contiguous subarray within a one-dimensional array of numbers.
- The algorithm runs in **O(n)** time, making it efficient for large datasets.

Why Use Kadane's Algorithm?

o Provides an optimal solution for maximum subarray sum problems.

o Useful in real-world scenarios like stock market analysis, where we want to maximize profit over a specific period.

Problem Statement

Given an array of integers, find the contiguous subarray (containing at least one number) which has the largest sum and return its sum.

Kadane's Algorithm Explanation

1. **Initialize variables**:
   - `max_current` = First element of the array.
   - `max_global` = First element of the array.
2. **Iterate through the array** starting from the second element:
   - Update `max_current` as:
     `max_current = max(current_element, max_current + current_element)`
   - Update `max_global` if `max_current` exceeds `max_global`:
     `max_global = max(max_global, max_current)`
3. **Return** `max_global` as the result, which holds the maximum subarray sum.

Algorithm Steps

1. Initialize:

   ```
   cpp
   Copy code
   int max_current = arr[0];
   int max_global = arr[0];
   ```

2. Loop through array:

```
cpp
Copy code
for(int i = 1; i < arr.size(); i++) {
    max_current = max(arr[i], max_current + arr[i]);
    if(max_current > max_global)
        max_global = max_current;
}
```

3. Output:
   - `max_global` is the maximum sum of the subarray.

Example

Input: `arr = [-2, 1, -3, 4, -1, 2, 1, -5, 4]`
Output: `6`

- Explanation: The maximum sum subarray is `[4, -1, 2, 1]` with sum `6`.

Home Task

1. **Implement Kadane's Algorithm in C++**
   - Write a function that takes an array as input and returns the maximum subarray sum.
2. **Modify Kadane's Algorithm**
   - Extend the algorithm to also return the start and end indices of the maximum subarray.

Conclusion

Kadane’s Algorithm is a cornerstone technique in dynamic programming. It demonstrates how to efficiently solve problems involving contiguous subarrays. Mastering Kadane's Algorithm will enhance your ability to tackle optimization problems involving arrays

---

Day 10 Notes

Prepared by Munawar Johar