Question 1: Prefix Sum Array Concept and Applications

Concept

A prefix sum array is a data structure that stores the running sum of elements in an array. For an array A, the prefix sum array P is defined such that P[i] = A[0] + A[1] + ... + A[i].

In other words, each element at index i in the prefix sum array represents the sum of all elements from index 0 to i in the original array.

Applications

1. **Range Sum Queries**: Quickly compute the sum of elements between any two indices in O(1) time after O(n) preprocessing.
2. **Equilibrium Point Detection**: Find points where the sum of elements to the left equals the sum of elements to the right.
3. **Subarray Sum Problems**: Efficiently find subarrays that sum to a given value or determine if such subarrays exist.
4. **2D Array Operations**: Can be extended to 2D prefix sums for efficient area calculations.
5. **Difference Arrays**: Used in conjunction with difference arrays to perform range updates efficiently.
6. **Sliding Window Optimizations**: Helps optimize certain sliding window computations.
7. **Pattern Matching**: Used in certain string and pattern matching algorithms.

Implementation

cpp

```cpp
vector<int> buildPrefixSum(const vector<int>& arr) {
    int n = arr.size();
    vector<int> prefixSum(n);

    prefixSum[0] = arr[0];
    for (int i = 1; i < n; i++) {
        prefixSum[i] = prefixSum[i-1] + arr[i];
    }

    return prefixSum;
}
```

Example

Consider array: [3, 1, 4, 8, 2]

The prefix sum array would be:

- P[0] = 3
- P[1] = 3 + 1 = 4
- P[2] = 4 + 4 = 8
- P[3] = 8 + 8 = 16
- P[4] = 16 + 2 = 18

Resulting prefix sum array: [3, 4, 8, 16, 18]

With this prefix sum array, we can find the sum of elements between any indices i and j (inclusive) using the formula:

- If i = 0: P[j]
- If i > 0: P[j] - P[i-1]

This gives us constant-time access to any range sum in the original array.