Prefix Sum Techniques

When? : When you are asked to query a range from l to r in an array multiple times.

Why? : To Improve efficiency of range queries from O(q*n) to O(q+n).

What? : Create a new array which will have a sum of prefixes till index i on ith index.

Ex - if array A = [1, 2, 3, 4, 5, 6]
Than prefix sum Array = [1, 3, 6, 10, 15, 21]

If range is l to r then result = prefix_sum[r] - prefix_sum[l].

Time Complexity : O(q+n) where q is no of queries and n is len A.

O(q+n) = [q in O(1)] & [n in O(n) for building prefix sum]

Limitations :

1. Array must be static and updates are not allowed.
2. Operation can only be performed if the operator is Invertible like (+ to -). Ex of Invertible - [+, -, ⊕] & Ex of Non Invertible : [min, max]
3. Commutativity : A (op) B = B (op) A // op = some operator. Ex - A+B = B+A. Commutativity is not compulsory to perform prefix sum technique. We can use this technique in matrix multiplication which is not commutative.