Prefix Sums Cheatsheet
(USACO Silver)

Use this algorithm when:

- You want to **get the sum of a range** in an array or you want to **count the number of something in a range** of an array
- The numbers in the array don't change

- OR -

- You have parenthesis

Steps to using the algorithm:

Thinking about the problem:

1) Make the array something we want to take prefix sums on
    - If we want the sum of a range, we don't need to change the array
    - If we want to count the number of something, make the thing we are counting a 1 and everything else a 0
    - If we have parenthesis, make '(' +1 and make ')' -1, take the prefix sums (do everything up to step 3), then skip step 4 and refer to the sort and sweep cheatsheet to complete the problem and for runtime. You likely have to sweep.
        - Usually refer to balanced parenthesis
        - As soon as the prefix sums become negative, the string is no longer balanced
        - If the prefix sum of the current index is 0 and the prefix sum was never negative before the current index, then it is balanced

Implementation:

2) Whenever you read input, convert to 1-based indexing
3) Create the prefix sum array
    ```
    for (int i=1; i<=n; i++) {
          prefix[i] = prefix[i-1] + value[i]
    }
    ```
4) For each query (if there are queries):
    - Solve the query in O(1) using the prefix sum array
      ```
      prefix[b] - prefix[a-1]
      ```

How to calculate the runtime:

O(N + Q)

Copyright © 2020 VPlanet Coding. All Rights Reserved.