Arrays Challenge-Subarray with given sum (Google, Amazon, Facebook, Visa)

Problem

Given an unsorted array **A** of size **N** of non-negative integers, find a continuous subarray which adds to a given number **S.**

Constraints

$1 <= N <= 10^5$

$0 <= Ai <= 10^{10}$

Example

Input**:**

N = 5, S = 12

A[] = {1,2,3,7,5}

Output**:** 2 4

Explanation**:** The sum of elements from 2nd position to 4th position is 12.

Solution

Brute Force Solution

- Find sum of all possible subarrays. If any of the sum equates to **S,** output the starting and ending index of the subarray.

Time Complexity : **O(n^2)**

Optimized Approach

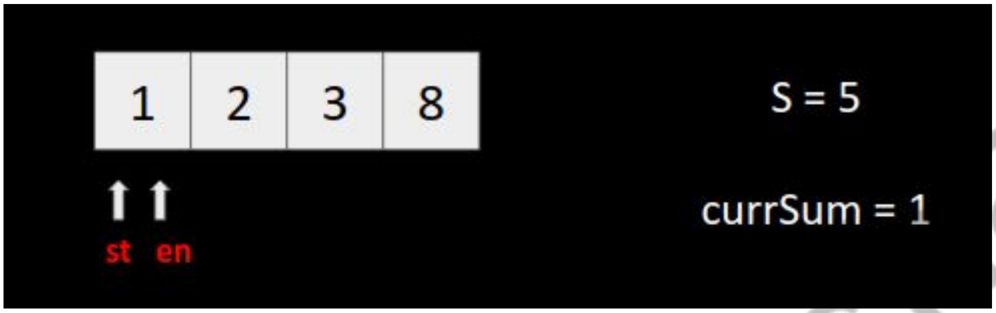
Steps:

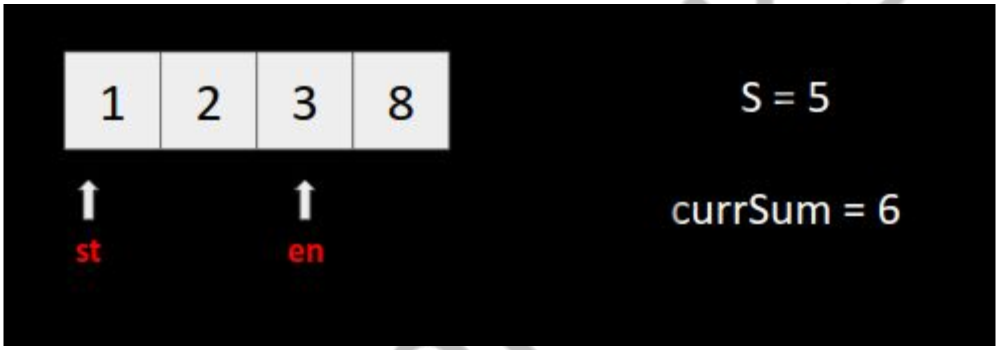
1. Keep the pointers st and en, and a variable currSum that stores the sum from st to en.
2. initialize st = 0, en = 0
3. Increment en till currSum + a[en + 1] > S
4. When 3rd condition occurs, start increasing st until currSum <= S.

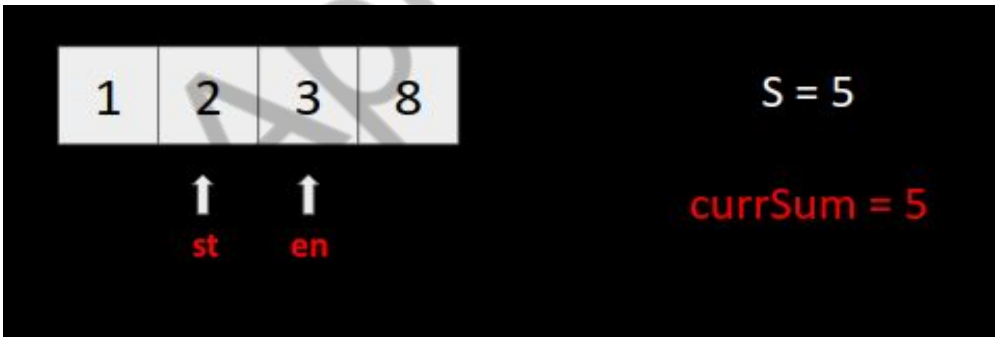
5. Whenever the condition (currSum = S) is satisfied, store st and en and BREAK from the loop.

Time Complexity: **O(n)**

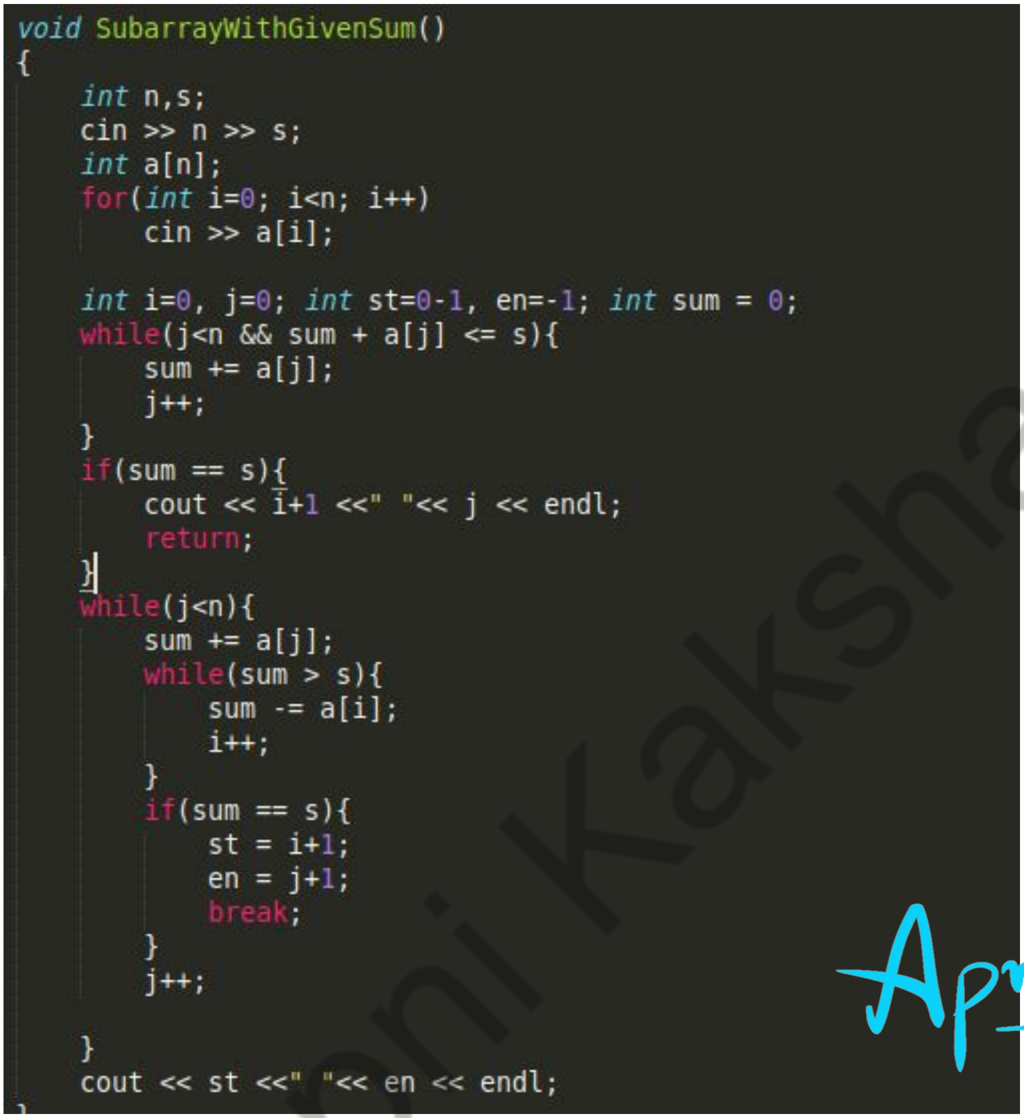
Iterations







Code:



```
void SubarrayWithGivenSum()
{
    int n,s;
    cin >> n >> s;
    int a[n];
    for(int i=0; i<n; i++)
        cin >> a[i];

    int i=0, j=0; int st=0-1, en=-1; int sum = 0;
    while(j<n && sum + a[j] <= s){
        sum += a[j];
        j++;
    }
    if(sum == s){
        cout << i+1 <<" "<< j << endl;
        return;
    }
    while(j<n){
        sum += a[j];
        while(sum > s){
            sum -= a[i];
            i++;
        }
        if(sum == s){
            st = i+1;
            en = j+1;
            break;
        }
        j++;

    }
    cout << st <<" "<< en << endl;
}
```