Data Structures and Algo in Java - Day 13

Learnt about Basic Sliding Window Algorithm for finding the Longest sub array which sums to k  
  
here is the Code

import java.util.\*;

public class LongestSubarraySumK

{

public static void main(String[] args) {

int arr[] = {1, 2, 3, 1, 1, 1, 1, 4, 2, 3};

int k = 6;

int ans1 = normal(arr,k);

int ans2 = optimized(arr,k);

System.out.println(ans1);

System.out.println(ans2);

}

public static int normal(int arr[],int k)

{

HashMap<Integer, Integer> map = new HashMap<>();

int sum = 0;

int maxLength = 0;

for (int i = 0; i < arr.length; i++)

{

sum += arr[i];

if (sum == k) {

maxLength = i + 1;

}

if (map.containsKey(sum - k)) {

int prevIndex = map.get(sum - k);

maxLength = Math.max(maxLength, i - prevIndex);

}

if (!map.containsKey(sum)) {

map.put(sum, i);

}

}

return maxLength;

}

public static int optimized(int arr[],int k)

{

int left = 0;

int right = 0;

int n = arr.length;

int sum = arr[0];

int maxlen=0;

while(right<n)

{

while(left<=right && sum>k)

{

sum = sum - arr[left];

left++;

}

if(sum == k)

{

maxlen=Math.max(maxlen,right-left+1);

}

right++;

if(right<n)

{

sum = sum+arr[right];

}

}

return maxlen;

}

}