

DSA Subarray Sum Solution

```
Desidn DSA notes along with this code: public int
subarraySum(int[] nums, int k) {
    int sum = 0, count = 0;
    Map<Integer, Integer> map = new HashMap<>();
    map.put(0, 1); // Initialize with one occurrence of sum = 0.

    for (int i = 0; i < nums.length; i++) {
        sum += nums[i];
        // Check if sum-k exists in the map.
        if (map.containsKey(sum - k)) {
            count += map.get(sum - k);
        }
        // Increment the count of the current sum in the map.
        map.put(sum, map.getOrDefault(sum, 0) + 1);
    }

    return count;
}
```

 Subarray Sum Notes

I've created detailed notes on the "Subarray Sum Equals K" problem, explaining the code, key concepts, and execution flow. Let me know if you need any further refinements or additions!