**Topic – HASHING**

**FORMAT : NAME, CODE LINK, HOW I SOLVED (DESC)**

**(Remember What Already occurred and How many Times).**

1. **Use Map For checking the freq of elements of before the current position while traversing.**

**TOPIC: Hashing**

**1Q) Sum[L………R]:**

**SolN: Prefix Sum-[L-1 to R].**

**DESC:**

**Pre[R] – Pre[l-1]……**

**PreCompute The values and Use.**

**2Q) Subarrays with Sum == K:**

**SolN: add freq of [sum-k] if present and each time freq[sum]++;**

**DESC: check sum-k present in map  
than add count….**

**3Q) Longest Consecutive Sequence:  
 SOln: Two HashMap’s for present and checked purpose  
 Desc: Use Two maps for present and checked.**

**4Q) Largest subarray with 0 sum**

**Soln: Check Number present two times and we get zer subarray between.  
Desc: Use map for checking subarray sum of array  
 (2…….2…)cnt = 7.**

**5Q) cnt pairs with xor == K  
 SolN: find curr^k find in map than cnt += mp[curr^k]**

**and mp[curr]++,if(curr == k)cnt++;**

**Desc : use map and cnt no of pairs.**

**Q) For some questions modify array such 0 🡪 -1.  
 and also initialize some maps with initial value of mp[0] = 0,  
 -1 or 1 based on Question.  
  
  
 Q)** [**2131. Longest Palindrome by Concatenating Two Letter Words**](https://leetcode.com/problems/longest-palindrome-by-concatenating-two-letter-words/)

**Q)**