**LeetCode Problems for Coding Interviews**

**1. Arrays & Hashing**

**Q.1. Two Sum**

Given an array of integers “nums” and an integer “target”, return indices of the two numbers such that they add up to “target”.

Input: nums = [2,7,11,15], target = 9

Output: [0,1]

Brute-force: Her eleman için diğer tüm elemanlarla toplamlarını kontrol ederiz. Yani tüm sayı çiftleri denenir. Bu yöntem küçük diziler için çalışır ancak büyük veri setlerinde performans sorunu yaşanır.

Time Complexity: **O(n2)**

Space Complexity: **O(1)**

**Optimal:** Bir “HashMap” yani bir anahtar-değer tablosu kullanarak sayıları işleriz. Her sayının hedefe ulaşmak için ihtiyaç duyduğu tamamlayıcı sayıyı daha önce görüp görmediğimizi kontrol ederiz. Eğer görmüşsek çözümü bulmuş oluruz.

Time Complexity: **O(n)**

Space Complexity: **O(n)**

Optimal Çözüm Java Kodu:

public int[] twoSum(int[] nums, int target) {  
 Map<Integer, Integer> map = new HashMap<>();  
  
 for (int i = 0; i < nums.length; i++) {  
 int complement = target - nums[i];  
  
 if (map.containsKey(complement)) {  
 return new int[] { map.get(complement), i };  
 }  
 map.put(nums[i], i);  
 }  
  
 return new int[] {};  
}