

# İki Sayının Toplamı Problemi

## Amaç

Verilen bir tamsayı dizisi içinde, belirli bir hedef toplamı oluşturan iki elemanın indekslerini bulmak ve farklı algoritmaların etkinliğini analiz etmek.

## Açıklama

Bu problem, algoritma ve veri yapıları alanında yaygın bir sorundur. İki farklı yaklaşımla çözülebilir:


1. **Brute Force (Kaba Kuvvet) Yöntemi:** Dizi içinde tüm olası ikilileri kontrol ederek hedef toplamı oluşturan çifti bulur. Zaman karmaşıklığı  $O(n^2)$  olur.
2. **HashMap Kullanarak Optimizasyon:** Bir hash tablosu oluşturularak her elemanın tamamlayıcısını kontrol eder. Zaman karmaşıklığı  $O(n)$  olur ve daha verimli bir yaklaşımdır.

## Yöntem

1. **Diziyi ve hedef toplamı tanımlama:** Kullanıcıdan bir tamsayı dizisi ve hedef toplam alınacak.
2. **Algoritma 1: Brute Force Yöntemi**
  - Dizi içinde tüm olası çiftler kontrol edilecek.
  - İki sayının toplamı hedefe ulaşıyorsa, bu sayıların indeksleri döndürülecek.
3. **Algoritma 2: HashMap Kullanarak Optimizasyon**
  - Dizi boyunca her eleman için tamamlayıcı (hedef – eleman) hash tablosunda aranacak.
  - Tamamlayıcı varsa, indeksler döndürülecek.
4. **Karşılaştırma:**
  - İki algoritmanın çalışma süresi ve etkinliği karşılaştırılacak.

## Örnekler


python

 Copy  Edit

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

Output: `[0, 1]` # Çünkü  $nums[0] + nums[1] = 2 + 7 = 9$



python

 Copy  Edit

Input: `nums = [3, 2, 4]`, `target = 6`

Output: `[1, 2]` # Çünkü  $nums[1] + nums[2] = 2 + 4 = 6$



python

 Copy  Edit

Input: `nums = [1, 5, 3, 8]`, `target = 9`

Output: `[0, 3]` # Çünkü  $nums[0] + nums[3] = 1 + 8 = 9$

python

 Copy  Edit

Input: `nums = [10, 20, 30, 40, 50]`, `target = 70`

Output: `[2, 3]` # Çünkü  $nums[2] + nums[3] = 30 + 40 = 70$