LaTex Template

Hao Yin

September 28, 2025

Question

Given an integer array `nums` and an integer `target`, find *two* distinct elements in the array whose sum equals `target`, and return their indices. You may assume there is exactly one valid answer for each input, and you may not use the same element twice. Return the answer in any order.

Examples

- **Input:** nums $= [2, 7, 11, 15]$, `target` $= 9$
  Output: `[0,1]`
  Explanation: `nums[0]` + `nums[1]` $= 2 + 7 = 9$.

- **Input:** nums $= [3, 2, 4]$, `target` $= 6$
  Output: `[1,2]`

- **Input:** nums $= [3, 3]$, `target` $= 6$
  Output: `[0,1]`

Constraints

$$2 \leq \texttt{nums.length} \leq 10^4,$$
$$-10^9 \leq \texttt{nums}[i] \leq 10^9,$$
$$-10^9 \leq \texttt{target} \leq 10^9,$$
$$\text{Exactly one valid answer exists.}$$

Follow-up Can you design an algorithm with time complexity strictly better than $O(n^2)$?

Solution

First attempt: use a sort to get a ordered array and two for loop and add from the left side to the right side. Will not work because after sorting the index is messed up, one way is to use if condition to match each element in origin array and ordered array but it can not work for same element with different index. (2025/9/27)

Brute Force: use the dumbest brute force solution to traverse all possible solutions. (Passed:91 ms, 13.83 MB 2025/9/27)

Hash Map: use hash map and check the complement, from $O(n^2) \rightarrow O(n)$. (Passed: 3 ms, 14.51 MB 2025/9/28)