Two Sum

Solving the Two Sum problem - a classic hashmap based question with an efficient linear-time solution.

Strategy: Use a Hash Map:

1. Traverse the array once.
2. For each nums[i], compute the complement: complement = target - nums[i]
3. Check if the complement already exists in the map:
   - If yes, return the indices: [map[complement], i]
   - Otherwise, store the current value in the map: map[nums[i]] = i;

Why it works?: You're essentially asking: "Have I already seen a number that would add with nums[i] to get target?"

Time and space complexity: - Time: O(n): - each element is processed once.
- Space: O(n): - worst-case storage in the map.

Edge Cases Handled:
- Duplicate values (like [3, 3])
- Negative numbers
- Exactly one valid solution, as guaranteed by the problem.