Problem Solving Quiz (Grade 10)

1. Count Equal Pairs

Description

Given an array of positive integers `nums`, count the number of pairs `(i,j)` where `nums[i] == nums[j]` and `i < j`. You must solve this problem using an **iterative approach** without recursion.

Example 1

```
Input: nums = [1,2,3,1,1,3]
Output: 4
Explanation: The pairs with equal values are:
- (0,3): nums[0] = 1, nums[3] = 1
- (0,4): nums[0] = 1, nums[4] = 1
- (3,4): nums[3] = 1, nums[4] = 1
- (2,5): nums[2] = 3, nums[5] = 3
```

Example 2

```
Input: nums = [1,1,1,1]
Output: 6
Explanation: All pairs (0,1), (0,2), (0,3), (1,2), (1,3), (2,3) have
    equal values.
```

Constraints

- $1 \leq$ nums.length ≤ 100
- $1 \leq$ nums[i] ≤ 100

2. Subset Sum Finder

Description

Given an array of positive integers `nums` and a positive integer `target`, return `true` if there exists a subset of `nums` that sums to `target`, and `false` otherwise. You must solve this problem using a **recursive approach**.

Example 1

```
Input: nums = [3,34,4,12,5,2], target = 9
Output: true
Explanation: The subset [4,5] sums to 9.
```

Example 2

```
Input: nums = [1,2,3], target = 7
Output: false
Explanation: No subset of [1,2,3] sums to 7.
```

Constraints

- $1 \leq$ nums.length ≤ 20
- $1 \leq$ nums[i], target $\leq 10^5$

3. Minimum Waiting Time

Description
Given an array of positive integers `queries`, where each element represents the execution time of a query, reorder the queries to minimize the total waiting time. The waiting time for a query is the sum of execution times of all queries executed before it. Return the minimum total waiting time. You must solve this problem using a **greedy approach**.

Example

```
Input: queries = [3,2,1,2,6]
Output: 17
Explanation: The optimal order is [1,2,2,3,6]. Waiting times:
- Query 1: 0 = 0
- Query 2: 1 = 1
- Query 3: 1+2 = 3
- Query 4: 1+2+2 = 5
- Query 5: 1+2+2+3 = 8
Total waiting time = 0 + 1 + 3 + 5 + 8 = 17.
```

Constraints

- $1 \leq$ queries.length $\leq 10^4$
- $1 \leq$ queries[i] $\leq 10^4$

4. Closest Pair of Points

Description
Given `n` points in a 2D plane, where each point is represented as an array `[x, y]` of integers, find the minimum Euclidean distance between any two distinct points. Return the distance rounded to 3 decimal places. You must solve this problem using a **divide and conquer** approach.

Example

```
Input: points = [[2,3],[12,30],[40,50],[5,1],[12,10],[3,4]]
Output: 1.414
Explanation: The closest pair is [2,3] and [3,4] with Euclidean
   distance sqrt((3-2)^2 + (4-3)^2) \approx 1.414.
```

Constraints

- $2 \leq$ points.length $\leq 10^4$
- $-10^5 \leq$ points[i][0], points[i][1] $\leq 10^5$
- No two points have the same coordinates.

5. Longest Increasing Subsequence

Description
Given an array of integers `nums`, return the length of the longest **strictly increasing** subsequence. A subsequence is a sequence that can be derived by deleting some or no elements without changing the order of the remaining elements. You must solve this problem using a **dynamic programming** approach.

Example 1

```
Input: nums = [10,9,2,5,3,7,101,18]
Output: 4
Explanation: The longest increasing subsequence is [2,3,7,101], with
    length 4.
```

Example 2

```
Input: nums = [0,1,0,3,2,3]
Output: 4
Explanation: The longest increasing subsequence is [0,1,2,3], with
   length 4.
```

Constraints

- $1 \leq$ nums.length ≤ 2500
- $-10^4 \leq$ nums[i] $\leq 10^4$