

1. Given an integer array `nums`, return `true` if any value appears at least twice in the array, and return `false` if every element is distinct.
2. You are given an $m \times n$ matrix `mat` and two integers `r` and `c` representing the number of rows and the number of columns of the wanted reshaped matrix.
The reshaped matrix should be filled with all the elements of the original matrix in the same row-traversing order as they were.
3. A phrase is a palindrome if, after converting all uppercase letters into lowercase letters and removing all non-alphanumeric characters, it reads the same forward and backward. Alphanumeric characters include letters and numbers. Given a string `s`, return `true` if it is a palindrome, or `false` otherwise.
4. Given two binary strings `a` and `b`, return their sum as a binary string.
5. The Hamming distance between two integers is the number of positions at which the corresponding bits are different. Given two integers `x` and `y`, return the Hamming distance between them.
6. Given a signed 32-bit integer `x`, return `x` with its digits reversed. If reversing `x` causes the value to go outside the signed 32-bit integer range $[-2^{31}, 2^{31} - 1]$, then return 0.
7. Write a function `validate_brackets(s)` that takes a string as input and checks whether the brackets in the string are balanced or not. The string may contain the brackets `()`, `[]`, `{}`, and any other characters. The function should return `True` if the brackets are balanced, and `False` otherwise. For example, if the input string is `"(a + [b * c]) - {d / e}"`, the function should return `True`.
8. Write a recursive function `count_items(lst)` that takes a list as input and returns the number of items in the list (including nested lists). For example, if the input list is `[1, [2, 3], [4, [5, 6]], 7]`, the function should return 7.
9. Write a recursive function `sum_digits(n)` that takes a non-negative integer `n` as input and returns the sum of its digits. For example, if the input value is 12345, the function should return 15 (i.e., $1 + 2 + 3 + 4 + 5$).
- 10.

Given two strings `word1` and `word2`, return the minimum number of operations required to convert `word1` to `word2`.

You have the following three operations permitted on a word:

- Insert a character
- Delete a character
- Replace a character

11. Given a `rows x cols` binary matrix filled with 0's and 1's, find the largest rectangle containing only 1's and return its area.
- 12.

A **path** in a binary tree is a sequence of nodes where each pair of adjacent nodes in the sequence has an edge connecting them. A node can only appear in the sequence **at most once**. Note that the path does not need to pass through the root.

The **path sum** of a path is the sum of the node's values in the path.

Given the `root` of a binary tree, return the maximum **path sum** of any **non-empty** path.

Given an input string (`s`) and a pattern (`p`), implement wildcard pattern matching with support for `'?'` and `'*'` where:

- 13.

- `'?'` Matches any single character.
- `'*'` Matches any sequence of characters (including the empty sequence).

The matching should cover the **entire** input string (not partial)

