Palindrome Partitioning Documentation

Objective

Given a string s, the objective is to partition s such that every substring of the partition is a palindrome. The function should return all possible palindrome partitionings of s.

Example 1:

- **Input:** s = "aab"
- **Output:** [["a","a","b"],["aa","b"]]
- **Explanation:** The string "aab" can be partitioned into palindromic substrings in two ways: ["a", "a", "b"] and ["aa", "b"].

Example 2:

- **Input:** s = "a"
- **Output:** [["a"]]
- **Explanation:** The string "a" is already a palindrome, so the only possible partition is ["a"].

Constraints

- The length of the string s is between 1 and 16, inclusive.
- The string s contains only lowercase English letters.

Approach

- The solution uses backtracking to explore all possible partitions of the string s and checks if each partition is a palindrome.

Steps

1. Helper Function (is_palindrome):

- This function checks if a given substring is a palindrome by comparing the substring with its reverse.

2. Backtracking Function (backtrack):

- This recursive function tries to partition the string starting from a given index.
- It iterates through possible end indices for the current substring.
- If the substring is a palindrome, it adds the substring to the current path and recursively calls itself with the next starting index.
- If the end of the string is reached, it adds the current path (which is a valid partition) to the result list.
- After exploring all possibilities for the current starting index, it removes the last added substring from the path to backtrack and try other possibilities.

3. Main Function (partition):

- Initializes the result list.
- Calls the backtracking function starting from index 0.
- Returns the result list containing all possible palindrome partitions.

Complexity

- The solution efficiently explores all possible partitions using backtracking, ensuring that each substring is checked only once for being a palindrome. This results in a time complexity of O(N * 2^N), where N is the length of the string. The space complexity is also O(N * 2^N) due to the storage of all possible partitions in the result list.