Problem 131: Palindrome Partitioning

Problem Information

Difficulty: Medium
Acceptance Rate: 73.23%
Paid Only: No
Tags: String, Dynamic Programming, Backtracking

Problem Description

Given a string `s`, partition `s` such that every substring of the partition is a **palindrome**. Return _all possible palindrome partitioning of_`s`.

Example 1:

Input: s = "aab" **Output:** [["a","a","b"],["aa","b"]]

Example 2:

Input: s = "a" **Output:** [["a"]]

Constraints:

* `1 <= s.length <= 16` * `s` contains only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
vector<vector<string>> partition(string s) {

}
};
```

Java:

```
class Solution {
public List<List<String>> partition(String s) {

}
}
```

Python3:

```
class Solution:
def partition(self, s: str) -> List[List[str]]:
```