131. Palindrome Partitioning

Medium

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"]]
```

General Approach

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

        def dfs(i):
            if i >= len(s):
                res.append(part.copy())
                return
            for j in range(i, len(s)):
                if self.isPali(s, i, j):
                    part.append(s[i : j + 1])
                    dfs(j + 1)
                    part.pop()

        dfs(0)
        return res

    def isPali(self, s, l, r):
        while l < r:
            if s[l] != s[r]:
                return False
            l, r = l + 1, r - 1
        return True
```