131. Palindrome Partitioning

题目描述:https://leetcode.com/problems/palindrome-partitioning/

> 给定一个字符串，将其分割成小的回文串。求分法。
> 例如:
>
> ```
> "aab" -> [["a","a","b"],["aa","b"]]
> "abc" -> [["a","b","c"]]
> ```

解题思路:

按照求subset的那种模式去算。

代码:

```
class Solution {
public:
    bool isPalindrome(const string& s, int start, int end) {
        while(start <= end) {
            if(s[start++] != s[end--])
                return false;
        }
        return true;
    }
    bool isP(const string &s, int len) {
        int i = 0;
        while(i<=len) {
            if(s[i++] != s[len--]) {
                return false;
            }
        }
        return true;
    }
    vector<vector<string> > findP(vector<vector<string> > &res, string s, vector<s
tring> &tmp, int pos) {
        if(pos == s.size()) {
            res.push_back(tmp);
        }
        for(int l = 1; pos+l-1 < s.size(); l++) {
            string sub = s.substr(pos, l);
            if(isPalindrome(s, pos, pos+l-1)) {
            //if(isP(sub, l-1)) {
                tmp.push_back(sub);
                findP(res, s, tmp, pos+l);
                tmp.pop_back();
            }
        }
        return res;
    }
    vector<vector<string>> partition(string s) {
        vector<vector<string> > res;
        vector<string> tmp;
        return findP(res, s, tmp, 0);
    }
};
```