132. Palindrome Partitioning II

QuestionEditorial Solution

My Submissions

- Total Accepted: **55513**
- Total Submissions: **246143**
- Difficulty: **Hard**

Given a string *s*, partition *s* such that every substring of the partition is a palindrome.

Return the minimum cuts needed for a palindrome partitioning of *s*.

For example, given *s* = "aab",

Return 1 since the palindrome partitioning ["aa","b"] could be produced using 1 cut.

Subscribe to see which companies asked this question

```
class Solution {

public:
bool checkPalindrome(string & s, int i, int j){
    while(i<j){
        if (s[i]!=s[j]) return false;
        i++;j--;
    }
    return true;
}
int minCount(string s,int k, vector<int> &hist) {
    if (k>=s.size()) return 0;
    if (k==s.size()-1) return 1;
    if (checkPalindrome(s,k,s.size()-1)) return 1;  // key 1, terminate in advance
    int minC = INT_MAX;
    for (int i=s.size()-1;i>=k;i--){   // important, check from long substr to short ones,
saving lots of time
        if (checkPalindrome(s,k,i)) {
            int temp = 1;
            if (hist[i+1]==INT_MAX) {
                hist[i+1]=minCount(s,i+1,hist);
            }
            temp+=hist[i+1];
            minC = minC<temp?minC:temp;
            if (minC==2) return minC;   // key 2, terminate in advance
        }
    }
    return minC;
}
int minCut(string s) {
    if (s.size()<=1) return 0;
    vector<int> hist(s.size()+1,INT_MAX);
    return minCount(s,0,hist)-1;
}
};
```