Problem Link:

<https://leetcode.com/problems/unique-length-3-palindromic-subsequences/>

Solution:

class Solution {

public:

int countPalindromicSubsequence(string s) {

unordered\_map<char, int> l;

unordered\_map<char, int> r;

unordered\_set<string> p;

int n = s.length();

for(char c : s)

{

r[c]++;

}

for(int j = 0; j < n; ++j)

{

char ch1 = s[j];

r[ch1]--;

for(auto& x : l)

{

char ch2 = x.first;

if(r[ch2] > 0)

{

p.insert({ch2, ch1, ch2});

}

}

l[ch1]++;

}

return p.size();

}

};