Lecture 8 Strings

1. [https://practice.geeksforgeeks.org/problems/palindrome-string0817/1/?category[]=Strings&category[]=Strings&problemStatus=solved&page=1&query=category[]StringsproblemStatussolvedpage1category[]Strings#](https://practice.geeksforgeeks.org/problems/palindrome-string0817/1/?category%5b%5d=Strings&category%5b%5d=Strings&problemStatus=solved&page=1&query=category%5b%5dStringsproblemStatussolvedpage1category%5b%5dStrings#)

class Solution{

public:

int isPlaindrome(string s)

{

int n = s.length();

int i = 0;

int j = n-1;

while(i<j)

{

if(s[i]==s[j])

{

i++;

j--;

}

else

{

return 0;

}

}

return 1;

}

};

1. <https://leetcode.com/problems/split-a-string-in-balanced-strings/>

class Solution {

public:

int balancedStringSplit(string s) {

int n = s.size();

int l = 0;

int r = 0;

int count = 0;

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

{

if(s[i]=='L')

{

l++;

}

else

{

r++;

}

if(l==r)

{

count++;

l = 0;

r = 0;

}

}

return count;

}

};

1. <https://leetcode.com/problems/reverse-words-in-a-string-iii/>

class Solution {

void swapping(string &s, int start, int end)

{

while(start < end)

{

swap(s[start], s[end]);

start++;

end--;

}

}

public:

string reverseWords(string s) {

int start = 0;

int end = 0;

while(s[end]!='\0')

{

if(s[end]==' ')

{

swapping(s, start, end-1);

start = end + 1;

}

end++;

}

swapping(s, start, end-1);

return s;

}

};

1. <https://leetcode.com/problems/reverse-words-in-a-string/>

class Solution {

public:

string reverseWords(string s) {

while(!s.empty() && s.front()==' ')

{

s.erase(s.begin());

}

while(!s.empty() && s.back()==' ')

{

s.pop\_back();

}

string temp;

string ans;

int n = s.size();

for(int i=n-1; i>=0; i--)

{

if(s[i]==' ' && s[i-1]==' ')

{

continue;

}

else if(s[i]!=' ')

{

temp = s[i] + temp;

}

else

{

ans = ans + temp + ' ';

temp = "";

}

}

ans = ans + temp;

return ans;

}

};