**完美的代价**

问题描述

　　回文串，是一种特殊的字符串，它从左往右读和从右往左读是一样的。小龙龙认为回文串才是完美的。现在给你一个串，它不一定是回文的，请你计算最少的交换次数使得该串变成一个完美的回文串。  
　　交换的定义是：交换两个相邻的字符  
　　例如mamad  
　　第一次交换 ad : mamda  
　　第二次交换 md : madma  
　　第三次交换 ma : madam (回文！完美！)

输入格式

　　第一行是一个整数N，表示接下来的字符串的长度(N <= 8000)  
　　第二行是一个字符串，长度为N.只包含小写字母

输出格式

　　如果可能，输出最少的交换次数。  
　　否则输出Impossible

样例输入

5  
mamad

样例输出

3

锦囊1

使用贪心算法。

锦囊2

从左到右枚举每个字符，移动对应字符。个数为单的字符放中间。

C++代码

#include<cstdio>

#include<iostream>

using namespace std;

int changes(char s[],char x,int n)

{

int i,change=0,j,k;

for(i=0;i<n/2;i++)

{

if(s[i]==x)

{

for(j=i;j<n-i-1;j++)

if(s[n-i-1]==s[j])

break;

change+=j-i;

for(k=j;k>i;k--)

s[k]=s[k-1];

s[i]=s[n-i-1];

}

else

{

for(j=n-i-1;j>=i;j--)

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

break;

change+=n-i-1-j;

for(k=j;k<n-i-1;k++) s[k]=s[k+1];

s[n-i-1]=s[i];

}

}

return change;

}

int main()

{

int n,i,k=0,b[26]={0},j;

char y,s[8001]={0};

scanf("%d\n",&n);

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

{

scanf("%c",&s[i]);

b[s[i]-'a']++;

}

char x;

for(j=0;j<26;j++)

{

if(b[j]%2!=0)

{

k++;

x=j+'a';

}

}

if(k>=2)

printf("Impossible\n");

else

{

printf("%d\n",changes(s,x,n));

return 0;

}

}

C代码

#include<stdio.h>

int changes(char s[],char x,int n);

char x='0';

int main(void)

{

int n,i,k=0,b[26]={0},j;

char y,s[8000]={0};

scanf("%d",&n);

getchar();

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

scanf("%c",&s[i]);

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

{

j=s[i]-'a';

b[j]++;

}

for(j=0;j<26;j++)

{

if(b[j]%2!=0)

{

k++;

x=j+'a';

}

}

if(k>=2)

printf("Impossible\n");

else

printf("%d\n",changes(s,x,n));

return 0;

}

int changes(char s[],char x,int n)

{

int i,change=0,j,k;

for(i=0;i<n/2;i++)

{

if(s[i]==x)

{

for(j=i;j<n-i-1;j++)

if(s[n-i-1]==s[j])

break;

change+=j-i;

for(k=j;k>i;k--)

s[k]=s[k-1];

s[i]=s[n-i-1];

}

else

{

for(j=n-i-1;j>=i;j--)

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

break;

change+=n-i-1-j;

for(k=j;k<n-i-1;k++)

s[k]=s[k+1];

s[n-i-1]=s[i];

}

}

return change;

}

Java代码

import java.io.\*;

public class Main{

public static void main(String args[]) throws IOException {

BufferedReader buf = new BufferedReader(

new InputStreamReader(System.in));

int n = Integer.parseInt(buf.readLine());

String s = buf.readLine();

char c[] = s.toCharArray();

int count = 0, num = 0;

char ch = '0';

int i, k = 0, j;

int b[] = new int[26];

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

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

j = c[i] - 'a';

b[j]++;

}

for (j = 0; j < 26; j++) {

if (b[j] % 2 != 0) {

k++;

ch = (char) (j + 'a');

}

}

if (k >= 2)

System.out.println("Impossible");

else

System.out.println(changes(c, ch, n));

}

public static int changes(char s[], char x, int n) {

int i, change = 0, j, k;

for (i = 0; i < n / 2; i++) {

if (s[i] == x) {

for (j = i; j < n - i - 1; j++)

if (s[n - i - 1] == s[j])

break;

change += j - i;

for (k = j; k > i; k--)

s[k] = s[k - 1];

s[i] = s[n - i - 1];

} else {

for (j = n - i - 1; j >= i; j--)

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

break;

change += n - i - 1 - j;

for (k = j; k < n - i - 1; k++)

s[k] = s[k + 1];

s[n - i - 1] = s[i];

}

}

return change;

}

}