字符串

1.字符串替换

2.最长公共前缀

3.罗马数组转换整数

4.删除连续字符串

5.寻找最小子字符串

题目：1.字符串替换

请实现一个函数，将一个字符串中的每个空格替换成“%20”。例如，当字符串为We Are Happy.则经过替换之后的字符串为We%20Are%20Happy。

public class Solution {

public String replaceSpace(StringBuffer str) {

int index=str.indexOf(" ");

while(index!=-1){

str=str.replace(index,index+1,"%20");

index=str.indexOf(" ");

}

return str.toString();

}

}

语法熟悉：

StringBuffer对象的函数

-----------------------------------------------------------------------

StringBuffer replace(int start,int end,String str);

字符串替换函数：[start,end) 从start开始，小于end的字符串替换成str

Int indexOf(String str)

定位str的第一个元素在字符串中第一次出现的位置。如果没有返回-1

Int lastIndexOf(String str)

定位str的第一个元素在字符串中最后出现的位置。

-----------------------------------------------------------------------

题目：2.最长公共前缀

// a b c d e f 顺序拿第一个字符串的每一个字符与剩下字符串 对应位置上做对比。

// a b c d

// a b

题目：3.罗马数组转换整数

题目: 4.删除连续字符串

语法熟悉:

输入一行,处理一行。

Scanner in =new Scanner(System.in);

while(in.hasNext()){

String str = in.nextLine();

}

题目：5.寻找最小子字符串

str1里找到包含独立小字符串str2中所有字符的最小子字符串str3。

需要注意：  
1、str1中有可能没有完整包含str2所有字符的情况，此时返回""，即为空字符串；  
2、str1不会为空，但str2有可能为空，此时返回整个str1；  
3、str2可能存在重复的字符，此时str3需要包含相等数量该字符；

"meituan2019","i2t"

"ituan2"

public class Solution {

public String getMinString (String str1, String str2) {

// write code here

if(str2==null||str2.length()==0){ return str1;}

int[] strA = new int[128];

int[] strB = new int[128];

int left = 0,right = 0;

for(int i=0;i<str1.length();i++){

strA[str1.charAt(i)]++;

}

for(int i=0;i<str2.length();i++){

strB[str2.charAt(i)]++;

}

// 没有完全包含所有字符,str2中有个字符多了几个或一个。

for(int k=0;k<str2.length();k++){//str.charAt()

if(strA[str2.charAt(k)]<strB[str2.charAt(k)]){

return "";

}

}

//find left

for(int k=0;k<str1.length();k++){

//保证相同数目的字符在范围内

if(--strA[str1.charAt(k)]<strB[str1.charAt(k)]){

left=k;

break;

}

}

for(int k=str1.length()-1;k>=0;k--){

if(--strA[str1.charAt(k)]<strB[str1.charAt(k)]){

right=k;

break;

}

}

return str1.substring(left,right+1);

}

}

既然是包含它的最小子序列，那么序列的左右两边必定包含str2的字符

1. 自己构建hash函数表，记录每个单词出现的次数。
2. 找出特殊情况，str2中某个字符的数量，少于str1中。
3. 找出left,与right

str1=aaabbc str2=aab [--3<2 index=0] [--2<2 index=1=left]

str1=abbcd str2=bbc [--1<0 index=0] [--2<2 index=1=left ]

left满足 str1当前字符在str2中，并且str1当前字符在右边的数量小于当前字符在str2中的数量. 即--strA[ str1.charAt(k) ]<strB[ str1.charAt(k) ]

hash表可以做的是：

1. 统计单词重复出现的次数。
2. 遍历字符串时候，通过确定的次数,来判定后续的字符中有多少个当前字符。