Given a string s, remove duplicate letters so that every letter appears once and only once. You must make sure your result is

**the smallest in lexicographical order**

 among all possible results.

**Example 1:**

**Input:** s = "bcabc"

**Output:** "abc"

**Example 2:**

**Input:** s = "cbacdcbc"

**Output:** "acdb"

**Constraints:**

* 1 <= s.length <= 104
* s consists of lowercase English letters.

class Solution {

public String removeDuplicateLetters(String s) {

int len=s.length();

Stack<Integer> noDupe=new Stack();

int lastIndx[]=new int[26];

boolean isSeen[]=new boolean[26];

for(int i=0;i<len;i++){

lastIndx[s.charAt(i)-'a']=i;

}

for(int i=0;i<len;i++){

int curr=s.charAt(i)-'a';

if(isSeen[curr]) continue;

while(!noDupe.isEmpty() && noDupe.peek()>curr && i<lastIndx[noDupe.peek()]){

isSeen[noDupe.pop()]=false;

}

noDupe.push(curr);

isSeen[curr]=true;

}

StringBuilder res=new StringBuilder();

while(!noDupe.isEmpty()){

res.append((char)(noDupe.pop()+'a'));

}

return res.reverse().toString();

}

}