**算法训练 字串统计**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个长度为n的字符串S，还有一个数字L，统计长度大于等于L的出现次数最多的子串（不同的出现可以相交），如果有多个，输出最长的，如果仍然有多个，输出第一次出现最早的。

输入格式

　　第一行一个数字L。  
　　第二行是字符串S。  
　　L大于0，且不超过S的长度。

输出格式

　　一行，题目要求的字符串。  
  
　　输入样例1：  
　　4  
　　bbaabbaaaaa  
  
　　输出样例1：  
　　bbaa  
  
　　输入样例2：  
　　2  
　　bbaabbaaaaa  
  
　　输出样例2：  
　　aa

数据规模和约定

　　n<=60  
　　S中所有字符都是小写英文字母。

提示

　　枚举所有可能的子串，统计出现次数，找出符合条件的那个

本题的C++参考代码如下：

#include<cstdio>

#include<cstring>

int main()

{

char s[65],str[65];

int max=0,t,n,len;

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

len=strlen(s);

if(n<=len)

{

char ss[65],tt[65];

for(int i=len;i>=n;i--)

{

ss[i]=tt[i]='\0';

for(int j=0;j<=len-i;j++)

{

t=1;

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

ss[k]=s[k+j];

for(int x=j+1;x<=len-i;x++)

{

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

tt[y]=s[y+x];

if(strcmp(ss,tt)==0)

t++;

}

if(t>max)

{

max=t;

strcpy(str,ss);

// printf("%s\n",str);

}

}

}

printf("%s\n",str);

}

}

本题的C参考代码如下：

#include<stdio.h>

#include<string.h>

int main(){

char S[1000],str[1000][1000],temp[100],out[100];

int L,i=0,s,otongji=0,ttongji,a,b,c;

scanf("%d%c%c",&L,&S[0],&S[0]);

while(S[i]!='\n'){

scanf("%c",&S[i+1]);

i++;

}

S[i] = '\0';

for(s=i+1;L<=s;L++){

for(a=0;a<s+1-L;a++){//赋值

for(b = 0;b < L;b++){

str[a][b]=S[a+b];

}

str[a][b] = '\0';

}

for(i = 0;i < a-1;){//比较

for(b = 0;b<a;b++){

if(str[b][0]!='\0'){

for(c = 0;c<L;c++){

temp[c]=str[b][c];

}

temp[c] = '\0';

ttongji = 1;

i++;

str[b][0] = '\0';

break;

}

}

for(b++;b<a;b++){

if(!strcmp(str[b],temp)){

ttongji++;

i++;

str[b][0] = '\0';

}

}

if(ttongji > otongji||(ttongji==otongji&&strlen(temp)>strlen(out))){

strcpy(out,temp);

otongji = ttongji;

}

}

}

i = 0;

while(out[i] != '\0'){

printf("%c",out[i]);

i++;

}

getchar();

return 0;

}

本题的Java参考代码如下：

import java.io.\*;

class Main

{

public static void main(String[] args) throws Exception

{

BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));

int A =Integer.parseInt(bf.readLine());

String s = bf.readLine().toLowerCase();

char c []=s.toCharArray();

int ok\_number = 0;

String ok\_s0="",ok\_s="";

if (A<=60 && A>0 && A<c.length){

char ok\_char[] =new char[c.length];

int max=1,m=1;

ok\_s=""+c[0];

ok\_char[A]='\0';

int a=0;int MAX=1;a=A;

for (a=A;a<c.length;a++)

{

for (int i=0;i<c.length-a ;i++ )

{

for (int j = i+1;j<=c.length-a ;j++ )

{

for (int k=0;k<a ;k++ )

{

if (c[i+k]==c[j+k])

{

ok\_char[ok\_number]=c[i+k];

ok\_number++;

}

}

if (ok\_number==a)

{

m++;

ok\_s0= new String(ok\_char,0,a);

}

ok\_number=0;

}

if (max<m)

{

max=m;

s = ok\_s0;

}

m=1;

}

if (MAX<=max)

{

MAX=max;

ok\_s=s;

}

max=1;

}

System.out.print(ok\_s);

}

}

}