**数据结构实验报告**

**实验四 病毒感染检测**

学 生 姓 名 ：杨仕振

学 号 ：2015301500349

二○一六年十一月

**一、简介**

【问题描述】研究者将人的DNA和病毒DNA均表示成由一些字母组成的字符串序列。根据人的DNA去判断是否感染病毒。（注意，人的DNA序列是线性的，而病毒的DNA序列是环状的)。

【基本要求】

采用字符串的模式匹配算法。

因为病毒DNA序列是环状的，存储病毒DNA序列的字符串长度扩大为2m，将病毒DNA序列连续存储两次。

-依次取得每个环状字符串，将此字符串作为模式串，将人的DNA序列作为主串，调用模式匹配算法进行模式匹配。

**二、算法说明**

首先手动输入所需的要的病毒的的DNA和人的DNA，将其存放在字符串结构中。

然后通过循环，产生病毒DNA环断裂可能产生的所有DNA，并将其人的DNA一一匹配，如果匹配成功，则显示感染的病毒，若始终未匹配成功，则证明没有感染病毒。

匹配算法采用了改进后的KMP算法。

**三、测试结果**

但是，测试结果未成功，一运行就崩溃了。所以后续有待改进。

**四、分析与探讨**

程序的主要空间复杂度源自于KMP算法，所以空间复杂度为O（N+M）；

而时间复杂度主要由感染检测函数的循环决定，故为O(N3).

程序存在的问题就是时间复杂度较大，运行相对较慢。

**附录：源代码**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define virusMaxsize 4//病毒的最大DNA长度

#define humanMaxsize 6//人的最大DNA长度

//备注：病毒DNA必须比人类DNA短

typedef struct {

char data[virusMaxsize + humanMaxsize];

int length;

}Sqstring;

void GetNextval(Sqstring \*&t, int nextval[])

{

int j = 0, k = -1;

nextval[0] = -1;

while (j<t->length)

{

if (k == -1 || t->data[j] != t->data[k])

{

j++; k++;

if (t->data[j] != t->data[k])

nextval[j] = k;

else

nextval[j] = nextval[k];

}

else

k = nextval[k];

}

}

int KMPIndex(Sqstring \*&s, Sqstring \*&t)

{

int nextval[virusMaxsize], i = 0, j = 0;

GetNextval(t, nextval);

while (i<s->length&&j<t->length)

{

if (j == -1 || s->data[i] == t->data[j])

{

i++;

j++;

}

else

j = nextval[j];

}

if (j >= t->length)

return(i - t->length);

else

return(-1);

}

void infectcheck(Sqstring \*&human,Sqstring \*&virus)

{

bool infected = false;

int n = 0, c = 0;

char viruslist[2\*virusMaxsize];

Sqstring newvirus;

Sqstring \*linevirus=(Sqstring \*)malloc(sizeof(Sqstring));

human->length = strlen(human->data);

virus->length = strlen(virus->data);

strcpy\_s(newvirus.data,virus->data);

strcat\_s(newvirus.data, virus->data);

newvirus.length = strlen(newvirus.data);

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

{

for (int j = i; j < virusMaxsize; j++)

{

linevirus->data[j-i] = newvirus.data[j];

linevirus->length = strlen(linevirus->data);

}

c = KMPIndex(human, linevirus);

}

if (c == -1)

{

printf("经检验该人感染病毒%s\n", virus->data);

infected = true;

}

if (infected == false)

printf("该人未感染病毒");

}

int main()

{

int a;

Sqstring \*human;

Sqstring \*virus;

human = (Sqstring \*)malloc(sizeof(Sqstring));

virus = (Sqstring \*)malloc(sizeof(Sqstring));

printf("病毒检测程序\n");

printf("------------------\n");

printf("请输入病毒的DNA序列,长度不超过%d,格式如：ab...\n", virusMaxsize - 1);

gets\_s(virus->data, virusMaxsize - 1);

printf("请输入人的DNA序列,不超过3个,每个长度不超过%d,格式如：abcab...\n", humanMaxsize - 1);

gets\_s(human->data, humanMaxsize - 1);

printf("下面开始检测");

infectcheck(human, virus);

system("PAUSE");

return 0;

}