**数据结构**

**——实验报告书**



姓名：樊雨晨

学号：201632110111

班级：软件工程(中外)16 班

学院：数理与信息工程学院

日期：2017年05月10日

# **实验 六 KMP算法**

## **问题**

给定一个源串s和n个子串stri。判断stri是否是s的子串。

## **实验思路**

编写求NEXT数组和INDEX的索引程序，将主串和模式串进行对比。

## **实验步骤**

I和j分别是主串和模式串的循环变量，在next数组中求解各个位置的值大小，通过比较T[i]和T[j]。

## **代码**

#**include**<iostream>

#**include**<string>

#**include**<vector>

**using** **namespace** std;

**void** **get\_nextval**(string &T, vector<**int**> &next) { // 算法4.8

// 求模式串T的next函数修正值并存入数组nextval。

**int** i = 0;

**int** j = -1;

next[0] = -1;

**while** (i<T.length()-1)

{

**if** (j==-1||T[i] == T[j])

{

++i; ++j;

**if** (T[i] != T[j])

{

next[i] = j;

}

**else**

{

next[i] = next[j];

}

}

**else**

{

j = next[j];

}

}

} // get\_nextval

**int** **Index\_KMP**(string &S, string &T, **int** pos)

{ // 算法4.6

// 利用模式串T的next函数求T在主串S中第pos个字符之后的位置的

// KMP算法。其中，T非空，1≤pos≤StrLength(S)。

vector <**int**> next(T.length());

**int** i = pos;

**int** j = 0;

get\_nextval(T,next);

**while** (i <(**int**)S.length() && j <(**int**)T.length())

{

**if** (j==-1||S[i] == T[j])

{ // 继续比较后继字符

++i;

++j;

}

**else**

{

j = next[j];

} // 模式串向右移动

}

**if** (j>=(**int**)T.length()) **return** i-(**int**)T.length(); // 匹配成功

**else** **return** 0;

} // Index\_KMP

**int** **main**()

{

string S,T;

cin >> S;

**int** n,i;

cin >> n;

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

{

cin >> T;

cout << (Index\_KMP(S, T, 0)?"yes":"no") << endl;

}

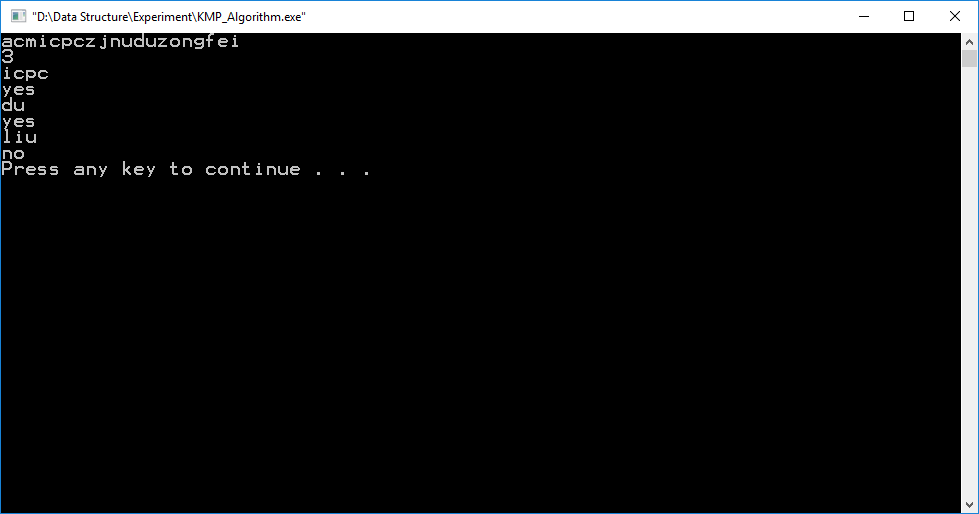
**return** 0;

}

## **测试**

正常，需要使用String类型表示字符串，NEXT需要使用Vector容器，这个是不定长的数组。不然需要使用大于10000的数组。

## **结果**



## **总结**

首先是要去看书明白NEXT数组的求解方法和匹配的方法，然后再具体代码的过程中，需要注意到由于测试的数据串长度比较长，所以要采用不定长的数组去运算。这次再次警示了我们要学习C++的方法去写程序。