

本科生课程论文(设计)

****

**题 目 数据结构**

学生姓名 沈宇豪

学 号 202083290386

学 院 计算机与软件学院

专 业 计算机类

授课教师 陈金辉

**二Ｏ二一 年 11 月 7 日**

目录

1. 需求分析-----------------------------------------------------------------------2
2. 概要设计-----------------------------------------------------------------------2
3. 详细设计-----------------------------------------------------------------------4
4. 调试分析-----------------------------------------------------------------------5
5. 附录代码-----------------------------------------------------------------------6

需求分析

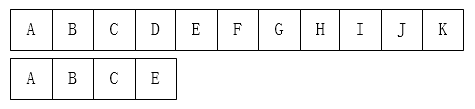
分为三个模块

1. 建立文本文件，实现文件名的输入
2. 进行字符串检索
3. 对检索到的字符串进行统计和输出

概要设计

主要为字符串匹配的算法设计，通常可以使用朴素模式的字符串匹配，但朴素模式的字符串匹配时间复杂度过大，针对此点可以使用KMP算法进行改进。

**KMP算法：**

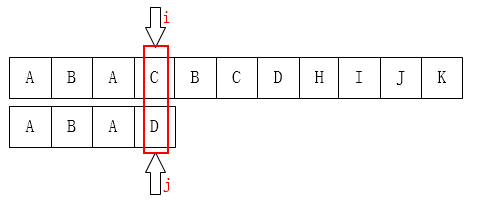


有两个字符串进行匹配

分别有指针i，j指向两个字符串。

对字符串进行从左到右的匹配

且保持i指针不回溯



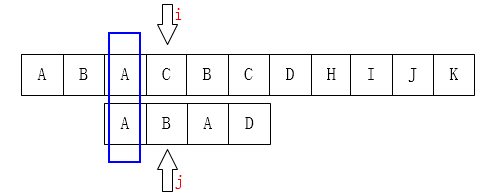
发现到第四个字符时，字符串不匹配，此时，将j（默认开始位置为第一个）开始向后移动k个位子（i保持不回溯）

最前面的q个字符和j之前的最后q个字符是一样的，或者可以说单看要比较的长的那个字符串中，最前面和最后面相比较，有q个字符是一样的。

K=q+1；

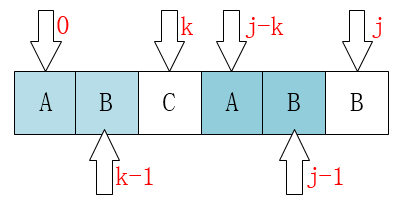
如此时，k=1+1=2

j就从第二个位子开始第二次匹配



由于数组存放

就有P[0 ~ k-1] == P[j-k ~ j-1]



设置一个next【】数组来给每一个位子存放他的k的值

当T[i] != P[j]时 有T[i-j ~ i-1] == P[0 ~ j-1]

P[0 ~ k-1] == P[j-k ~ j-1]

T[i-k ~ i-1] == P[0 ~ k-1]

**next[j] = k，表示当T[i] != P[j]时，j指针的下一个位置。**

**详细设计**

**int KMP(string line, string word, int flag = 0, int o = 0) {**

**int F[20], count = 0; //F[]就是next[]**

**F[0] = -1;**

**for (int i = 1; i < int(word.length()); i++) {**

**int j = F[i - 1];**

**while ((word[j + 1] != word[i]) && (j >= 0))**

**j = F[j];**

**if (word[j + 1] == word[i])**

**F[i] = j + 1;**

**else**

**F[i] = -1;**

**}**

**int i = 0, j = 0, num = 1, tflag = 0;**

**while (i < int(line.length())) {**

**if (line[i] == word[j]) {**

**i++;**

**j++;**

**if (j == int(word.length())) {**

**if (line[i]!=' '&&i!=int(line.length()))**

**{**

**j=0;**

**while(line[i]!=' ')**

**i++;**

**num++;**

**continue;**

**}**

**if (flag) {**

**if (!tflag){**

**cout << "第" << o << "行的：";**

**tflag=1;**

**}**

**cout << "第" << num << "个单词，";**

**}**

**count++;**

**j = F[j - 1] + 1;**

**}**

**} else {**

**if (j == 0)**

**i++;**

**else**

**j = F[j - 1] + 1;**

**}**

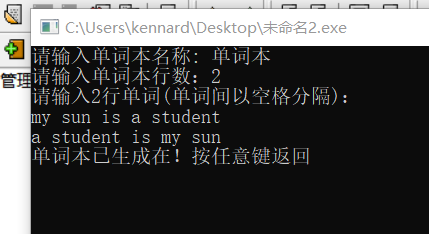
**if (line[i] == ' ') num++;**

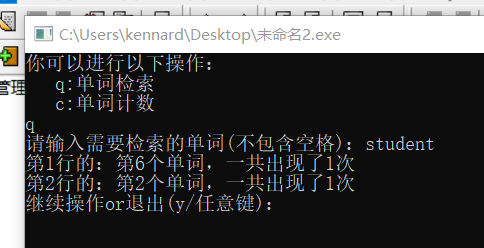
**}**

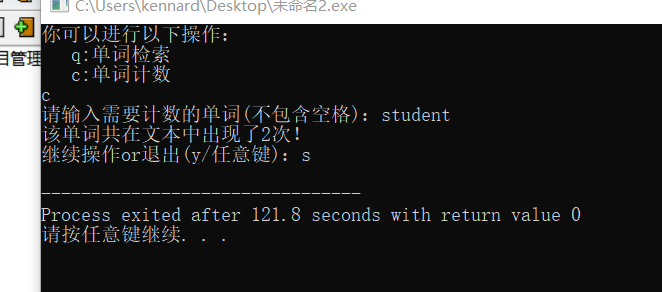
**return count;**

**}**

**调试分析**

****

****

****

**附录：**

**#include <bits/stdc++.h>**

**using namespace std;**

**string file\_name, a;**

**int KMP(string line, string word, int flag = 0, int o = 0) {**

**int F[20], count = 0; //F[]就是next[]**

**F[0] = -1;**

**for (int i = 1; i < int(word.length()); i++) {**

**int j = F[i - 1];**

**while ((word[j + 1] != word[i]) && (j >= 0))**

**j = F[j];**

**if (word[j + 1] == word[i])**

**F[i] = j + 1;**

**else**

**F[i] = -1;**

**}**

**int i = 0, j = 0, num = 1, tflag = 0;**

**while (i < int(line.length())) {**

**if (line[i] == word[j]) {**

**i++;**

**j++;**

**if (j == int(word.length())) {**

**if (line[i]!=' '&&i!=int(line.length()))**

**{**

**j=0;**

**while(line[i]!=' ')**

**i++;**

**num++;**

**continue;**

**}**

**if (flag) {**

**if (!tflag){**

**cout << "第" << o << "行的：";**

**tflag=1;**

**}**

**cout << "第" << num << "个单词，";**

**}**

**count++;**

**j = F[j - 1] + 1;**

**}**

**} else {**

**if (j == 0)**

**i++;**

**else**

**j = F[j - 1] + 1;**

**}**

**if (line[i] == ' ') num++;**

**}**

**return count;**

**}**

**void writeFile() {**

**system("cls");**

**int n;**

**cout << "请输入名称: ";**

**cin >> file\_name;**

**cout << "请输入行数：";**

**cin >> n;**

**ofstream outfile;**

**outfile.open((file\_name + ".txt").c\_str());**

**cout << "请输入"<<n<<"行单词(单词间以空格分隔)：" << endl;**

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

**getline(cin, a);**

**if (i==0) continue;**

**outfile << a << endl;**

**}**

**outfile.close();**

**cout<<"成功！按任意键返回"<<endl;**

**char c;**

**cin>>c;**

**}**

**void search(string word) {**

**ifstream infile;**

**int flag=0;**

**int count = 0, num = 1;**

**infile.open((file\_name + ".txt").c\_str());;**

**while (getline(infile, a)) {**

**count = KMP(a, word, 1, num);**

**if (count) {**

**cout << "一共出现了" << count << "次" << endl;**

**flag=1;**

**}**

**num++;**

**}**

**if (!flag) cout<<"该单词未在文本中出现！"<<endl;**

**}**

**void countNum(string word) {**

**int count = 0;**

**ifstream infile;**

**infile.open((file\_name + ".txt").c\_str());;**

**while (getline(infile, a)) {**

**count += KMP(a, word);**

**}**

**cout << "该单词共在文本中出现了" << count << "次！" << endl;**

**}**

**int main(int argc, char const \*argv[]) {**

**cout<<"目前无单词本，请先制作！继续或退出(y/任意键)"<<endl;**

**char c;**

**cin>>c;**

**if(c=='y') writeFile();**

**while(c=='y'){**

**system("cls");**

**cout<<"你可以进行以下操作："<<endl;**

**cout<<" q:单词检索"<<endl;**

**cout<<" c:单词计数"<<endl;**

**cin>>c;**

**if(c=='q'){**

**string word;**

**cout<<"请输入需要检索的单词(不包含空格)：";**

**cin>>word;**

**search(word);**

**}else if(c=='c'){**

**string word;**

**cout<<"请输入需要计数的单词(不包含空格)：";**

**cin>>word;**

**countNum(word);**

**}else{**

**cout<<"未包含该选项，重新选择or退出(y/任意键)：";**

**cin>>c;**

**continue;**

**}**

**cout<<"继续操作or退出(y/任意键)：";**

**cin>>c;**

**}**

**return 0;**

**}**