KMP算法

算法描述

KMP算法是一种字符串匹配算法，其基本思想是通过不断比较模式串与主串的对应字符，从而确定模式串的起始位置。

KMP算法的基本思想是：在模式串中，如果存在某个字符与主串的对应字符不匹配，则可以根据模式串的前缀来预测主串的下一个位置，从而减少比较次数。

KMP算法的具体步骤如下：

1. 首先，构造一个next数组，其长度为模式串的长度，next[i]表示模式串的前缀与后缀的最长相同的长度。
2. 然后，从模式串的第二个字符开始，逐个比较模式串与主串的对应字符。
3. 如果模式串的当前字符与主串的对应字符匹配，则继续比较下一个字符。
4. 如果模式串的当前字符与主串的对应字符不匹配，则根据next数组，预测主串的下一个位置。
5. 如果next[i]不等于0，则说明模式串的前缀与后缀的最长相同的长度为next[i]，则可以将模式串的比较位置向后移动next[i]个字符。
6. 如果next[i]等于0，则说明模式串的前缀与后缀没有相同的部分，则需要将模式串的比较位置向后移动1个字符。
7. 重复步骤3-6，直到模式串的末尾。

nextval的改进

在KMP算法中，使用next数组可以减少比较次数，但有时仍然会出现无谓的比较。为此，引入nextval数组。nextval数组是在next数组的基础上进行改进，使得在不匹配时，能够更进一步减少比较。

nextval[i]的定义如下：

- 如果模式串的前缀和后缀都有相同部分，此时nextval[i]的值为next[i]。
- 如果模式串的前缀与后缀没有相同部分，nextval[i]的值为下一个可能的匹配位置。

算法实现

C语言实现

1. 这是以 next[0]=0 来写的

```
void get_Next(string s, int next[])  //这个函数对字符串s进行预处理得到next数组
{
    int j = 0;
    next[0] = 0; //初始化
    for(int i = 1; i<s.size(); i++){ //i指针指向的是后缀末尾，j指针指向的是前缀末尾
        while(j>0&&s[i]!=s[j]) j = next[j-1]; //前后缀不相同，去找j前一位的最长相等前后缀
        if(s[i]==s[j]) j++; //前后缀相同，j指针后移
        next[i] = j; //更新next数组
    }
}
int strSTR(string s, string t) //这个函数是从s中找到t，如果存在返回t出现的位置，如果不存在返回-
{
    if(t.size()==0) return 0;
    get_Next(t, next);
    int j = 0;
    for(int i = 0; i < s.size(); i++){
        while(j>0&&s[i]!= t[j]) j = next[j-1];
        if(s[i]==t[j]) j++;
        if(j==t.size()) return i - t.size() + 1;
    }
    return -1;
}
```

2. 这是以 next[0]=-1 来写的

```
void get_Next(string s, int next[])  //这个函数对字符串s进行预处理得到next数组
{
    int j = -1;
    next[0] = -1; //初始化
    for(int i = 1; i<s.size(); i++){ //i指针指向的是后缀末尾，j指针指向的是前缀末尾
        while(j!=-1&&s[i]!=s[j+1]) j = next[j];
        if(s[i]==s[j+1]) j++; //前后缀相同，j指针后移
        next[i] = j; //更新next数组
    }
}
int strSTR(string s, string t) //这个函数是从s中找到t，如果存在返回t出现的位置，如果不存在返回-
{
    if(t.size()==0) return 0;
    get_Next(t, next);
    int j = -1;
    for(int i = 0; i < s.size(); i++){
        while(j>0&&s[i]!= t[j+1]) j = next[j];
        if(s[i]==t[j+1]) j++;
        if(j==t.size()-1) return i - j; // i - t.size() + 1;
    }
    return -1;
}
```

@title: KMP
@date: 2025-01-08 19:00:00
@version: 1.0.0

@author：软件工程宋浩元