**串的模式匹配方法**

需要用到的主要函数：

void StrInsert(SString& S, int pos, SString T) { //插入函数

int s\_len = StrLength(S);

int t\_len = StrLength(T);

if (s\_len + t\_len <= MAXSTRLEN) {

//先腾出空间，移动数据

t\_len = StrLength(T);

/\*for (int i = s\_len - 1; i >= pos; i--) {

S[i + t\_len] = S[i];

}

int j = pos;

for (int i = 0; i < t\_len; ++i) {

S[j + i] = T[i];

}

S[s\_len + t\_len] = '\0';\*/

}

else if (s\_len < MAXSTRLEN) {

t\_len = MAXSTRLEN - s\_len; //重新修改t\_len

/\*for (int i = s\_len - 1; i >= pos; i--) {

S[i + t\_len] = S[i];

}

int j = pos;

for (int i = 0; i < t\_len; ++i) {

S[j + i] = T[i];

}

S[s\_len + t\_len] = '\0';\*/

}

for (int i = s\_len - 1; i >= pos; i--) {

S[i + t\_len] = S[i];

}

int j = pos;

for (int i = 0; i < t\_len; ++i) {

S[j + i] = T[i];

}

S[s\_len + t\_len] = '\0';

}

void StrDelete(SString S, int pos, int len) { //删除函数

int s\_len = StrLength(S);

for (int i = pos; i < s\_len; i++) {

S[i] = S[i + len];

}

S[s\_len - len] = '\0';

}

模式匹配方法的代码：

**//最基本，最严格的方法 后续会有改进KMP**

int StrIndex(SString &S, SString T,int pos) {

int i = pos; int j = 0;

while (S[i] != '\0' && T[j] != '\0') {

if (S[i] == T[j]) {

i++;

j++;

}

else {

i = i-j+1; //这个式子是通过画图找规律得到的

j = 0;

}

}

if (T[j] == '\0') { //说明成功匹配，是因为找到了T的\0才退出了循环

return i - j;

}

else

return -1;

}

**//用V替换主串S中所有与T相等的不重叠的子串**

void Replace(SString &S, SString T, SString V) {

int s\_len = StrLength(S);

int t\_len = StrLength(T);

int v\_len = StrLength(V);

int index = -1;

int pos = 0;

while (pos < s\_len) {

index = StrIndex(S, T, pos);

if (index == -1)

return;

StrDelete(S, index, t\_len);

StrInsert(S, index, V);

pos = index + v\_len;

}

}

void main() {

SString S;

SString T;

SString V;

InitString(S);

InitString(T);

InitString(V);

StrAssign(S, "ababcababcab");

StrAssign(T, "abc");

StrAssign(V, "xy");

//模式匹配一

int index = StrIndex(S,T,0);

printf("%d", index);

//模式匹配二

Replace(S,T,V);

PrintString(S);

}