4.1 串类型的定义如下：

ADT String {

数据对象：

$D = \{ a_i \mid a_i \in CharacterSet，\ i=1,2,...,n,\quad n \geq 0 \}$

n称为串的长度

数据关系：

$R_1 = \{ < a_{i-1}, a_i > \mid a_{i-1}, a_i \in D，\quad i=2,...,n \}$

串（字符串）是由零个或多个字符组成的有限序列。由一对单引号相括，如: 'a string'

基本操作：

StrAssign (&T, chars)

DestroyString(&S)

StrCopy (&T, S)

StrLength(S)

StrCompare (S, T)

Concat (&T, S1, S2)

StrEmpty (S)

SubString (**&**Sub, S, pos, len)

ClearString (**&**S)

Index (S, T, pos)

Replace (**&**S, T, V)

StrInsert (**&**S, pos, T)

StrDelete (**&**S, pos, len)

} ADT String

StrAssign (&T, chars)

初始条件：chars 是字符串常量。

操作结果：把 chars 赋为 T 的值。

StrCopy (&T, S)

初始条件：串 S 存在。

操作结果：由串 S 复制得串 T。

StrEmpty(S)

初始条件：串S存在。

操作结果：若 S 为空串，则返回TRUE，否则返回 FALSE。

‘’ 表示空串，空串的长度为零。

DestroyString (&S)

初始条件：串 S 存在。

操作结果：串 S 被销毁。

StrCompare (S, T)

初始条件：串 S 和 T 存在。

操作结果：若S > T，则返回值 > 0；

若S = T，则返回值 = 0；

若S < T，则返回值 < 0。

例如：StrCompare（‘data’，‘state’）< 0

StrCompare（‘cat’，‘case’）> 0

StrLength (S)

初始条件：串 S 存在。

操作结果：返回 S 的元素个数，称为串的长度。

Concat (&T, S1, S2)

初始条件：串 S1 和 S2 存在。

操作结果：用 T 返回由 S1 和 S2联接而成的新串。

例如： Concate(T, ‘man’ , ‘kind’)

求得 T = ‘mankind’

SubString (**&**Sub, S, pos, len)

初始条件：

串 S 存在，**1≤pos≤StrLength(S)**

且**0≤len≤StrLength(S)-pos+1**。

操作结果：

用 Sub 返回串 S 的第 pos 个字符起长度为 len 的子串。

子串为“串” 中的一个字符子序列

例如： SubString(sub, ‘commander’ , 4, 3)

求得 sub = ‘man’ ;

SubString(sub, ‘commander’ , 1, 9)

求得 sub = ‘commander’ ;

SubString(sub, ‘commander’ , 9, 1)

求得 sub = ‘r’ ;

SubString(sub, ‘commander’ , 4, 7) sub = ?

SubString(sub, ‘beijing’ , 7, 2) sub = ?

起始位置和子串长度之间存在约束关系

SubString(‘student’ , 5, 0) = ‘’

长度为 0 的子串为“合法”串

Index (S, T, pos)

初始条件：串S和T存在，T是非空串，

1≤pos≤StrLength(S)。

操作结果：若主串 S 中存在和串 T 值相同的子串, 则返回它在主串 S 中第pos个字符之后第一次出现的位置；否则函数值为0。

“**子串在主串中的位置**”意指子串中的第一个字符在主串中的**位序**。

假设 S = ‘abcaabcaaabc’，T = ‘bca’

Index(S, T, **1**) = 2；　　Index(S, T, **3**) = 6；

Index(S, T, **8**) = 0；

Replace (&S, T, V)

初始条件：串S, T和 V 均已存在，且 T 是非空串。

操作结果：用V替换主串S中出现的所有与（模式串）T 相等的**不重叠**的子串。

例如：　假设 S = ‘abcaabcaaabca’ ，T = ‘bca’

若 V = ‘x’ ，则经置换后得到

S = ‘axaxaax’

若 V = ‘bc’ ，则经置换后得到

S = ‘abcabcaabc’

StrInsert (**&**S, pos, T)

初始条件：串S和T存在，1≤pos≤StrLength(S)＋1。

操作结果：在串S的第pos个字符之前插入串T。

例如：S = ‘chater’ ，T = ‘rac’ ，

则执行 StrInsert(S, 4, T) 之后得到

S = ‘character’

StrDelete (**&**S, pos, len)

初始条件：串S存在，1≤pos≤StrLength(S)-len+1。

操作结果：从串S中删除第pos个字符起，长度为len的子串。

ClearString (&S)

初始条件：串S存在。

操作结果：将S清为空串。

对于串的基本操作集可以有不同的定义方法，在使用高级程序设计语言中的串类型时，应**以该语言的参考手册为准。**

例如：C语言函数库中提供下列串处理函数：

gets(str)　　输入一个串；

puts(str)　输出一个串；

strcat(str1, str2)　串联接函数；

strcpy(str1, str2, k)　串复制函数；

strcmp(str1, str2)　串比较函数；

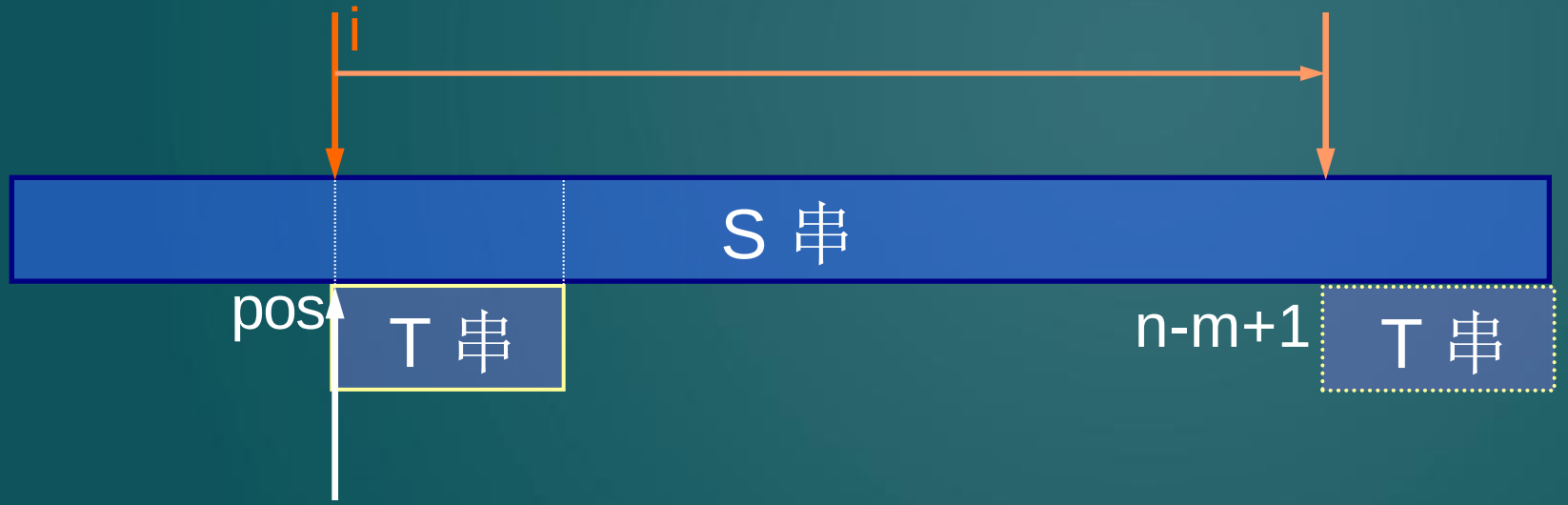
strlen(str)　　求串长函数；

在上述抽象数据类型定义的13种操作中， **串赋值StrAssign、串复制Strcopy、 串比较StrCompare、求串长StrLength、 串联接Concat以及求子串SubString 等六种操作构成串类型的最小操作子集。** 即：这些操作不可能利用其他串操作来实现， 反之，其他串操作（除串清除ClearString和串 销毁DestroyString外）可在这个最小操作子 集上实现。

例如，可利用串比较、求串长和求子串等操作实现定位函数Index(S,T,pos)。

算法的基本思想为：

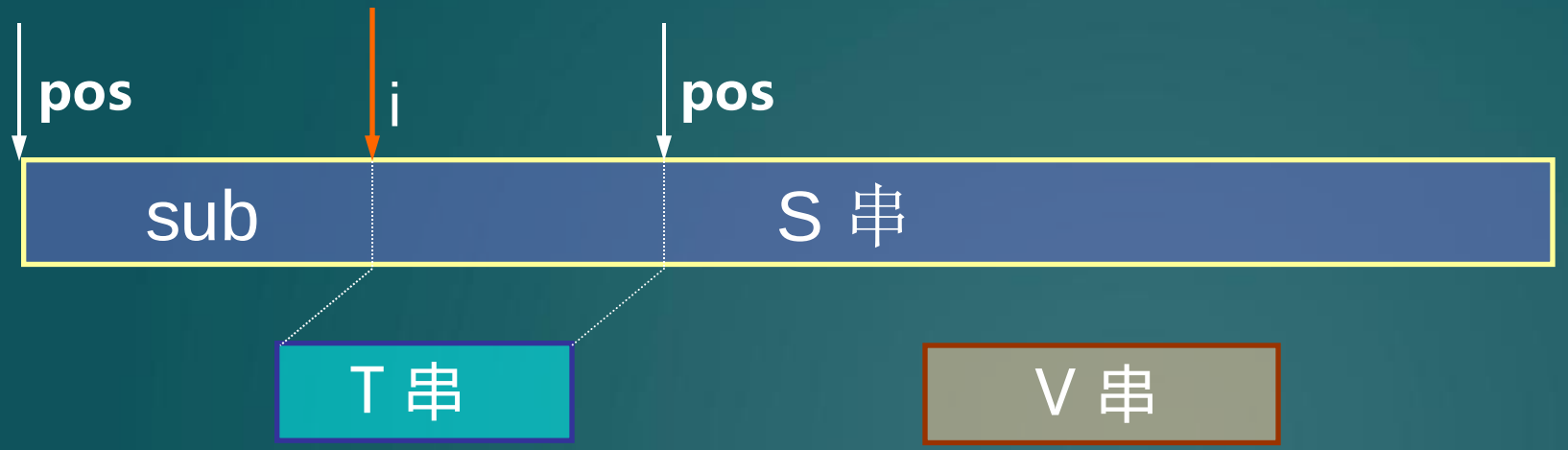
StrCompare(SubString(S, i, StrLength(T)),T) = 0 ?



```
int Index (String S, String T, int pos) {
 // T为非空串。若主串S中第pos个字符之后存在与 T相等的子串，
 //则返回第一个这样的子串在S中的位置，否则返回0
  if (pos > 0) {
    n = StrLength(S);  m = StrLength(T);  i = pos;
    while ( i <= n-m+1) {
        SubString (sub, S, i, m);
        if (StrCompare(sub,T) != 0)    ++i ;
        else return i ;
    } // while
  } // if
  return 0;          // S中不存在与T相等的子串
} // Index
```

算法：4.1

又如串的置换函数： **Replace** (&S, T, V)



news 串

串的逻辑结构和线性表极为相似，**区别**仅在于串的数据对象约束为字符集。

串的基本操作和线性表有很大差别。

在线性表的基本操作中，大多以“单个元素”作为操作对象； 在串的基本操作中，通常以“串的整体”作为操作对象。