

字符串:内容受限的线性表

- 1 空串:"" 空格串:" "
- 2 字符:字符串的基本单位,取值于字符集(二进制集,DNA 字符集,英语字母集)
- 3 编码:char->8bits,GB2312->两个字节,UNICODE
- 4 偏序编码->便于比较运算,大多数情况下为字典序,中文有笔划序
- 5 子串(个数为 $n(n-1)/2+1$)/真子串($n(n-1)/2-1$)
- 6 char s[m]长度固定,不是面向对象的数据类型, class String 采用动态变长的存储结构
- 7 静态存储(顺序存储):采用 char s[m]形式定义字符串,m 是常量保持不变,串的结束记'\0'
- 8 长度为 m 的字符串实际最大容量为(m - 1)
- 9 标准字符串函数:strlen(), strcpy(dest, src), strcat(dest, src), strcmp(), (左)定位函数 strchr(char* s, char c), 右定位函数 strrchr(char*s, char c)
- 10 字符串类:适应长度动态变化,采用动态变长,不合适采用动态链表
- 11 String 类算子:substr(), copy(), assign(), insert(), append(), +, find(), replace().....
- 12 String 创建算子:计算长度,开辟长度+1 的空间,strcpy()
- 13 抽取子串:从 index 开始,长度为 count
- 14 模式匹配:搜索目标字符串中与模板完全相同的子串,返回首字符位置
- 15 精确匹配(单选/多选(带通配符的匹配)/正则表达式) 近似匹配(字符串编辑距离)
- 16 朴素匹配:时间复杂度 $O(n * m)$
- 17 KMP 算法:模板中 每一个字符对应一个仅依赖于模式 P 的移位值
- 18 求移位值: $n[0] = -1; q[i] = q[k] \implies n[i+1] = k+1$, 否则 $k = n[k]$, 找不到 $k \implies n[i+1] = 0$
- 19 优化 $\implies q[i+1] = q[k]$ 时再优化
- 20 KMP 算法 \implies 看书/ppt
- 21 时间复杂度:分析 j++的执行次数,逼出 i++,j = n[i]的执行次数