

串是什么，串存储结构及其实现（无师自通）

数据结构中，字符串要单独用一种存储结构来存储，称为串存储结构。这里的串指的就是字符串。

严格意义上讲，串存储结构也是一种线性存储结构，因为字符串中的字符之间也具有"一对一"的逻辑关系。只不过，与之前所学的线性存储结构不同，串结构只用于存储字符类型的数据。

无论学习哪种编程语言，操作最多的总是字符串。数据结构中，根据串中存储字符的数量及特点，对一些特殊的串进行了命名，比如说：

- 空串：存储 0 个字符的串，例如 `S = ""`（双引号紧挨着）；
- 空格串：只包含空格字符的串，例如 `S = " "`（双引号包含 5 个空格）；
- 子串和主串：假设有两个串 a 和 b，如果 a 中可以找到几个连续字符组成的串与 b 完全相同，则称 a 是 b 的主串，b 是 a 的子串。例如，若 `a = "shujujiegou"`，`b = "shuju"`，由于 a 中也包含 "shuju"，因此串 a 和串 b 是主串和子串的关系；

需要注意的是，空格串和空串不同，空格串中含有字符，只是都是空格而已。另外，只有串 b 整体出现在串 a 中，才能说 b 是 a 的子串，比如 "shujiejougou" 和 "shuju" 就不是主串和子串的关系。

另外，对于具有主串和子串关系的两个串，通常会让你用算法找到子串在主串的位置。子串在主串中的位置，指的是子串首个字符在主串中的位置。

例如，串 `a = "shujujiegou"`，串 `b = "jiegou"`，通过观察，可以判断 a 和 b 是主串和子串的关系，同时子串 b 位于主串 a 中第 6 的位置，因为在串 a 中，串 b 首字符 'j' 的位置是 6。

本章，我们会学习两种模式匹配算法专门解决此类问题。

串存储结构的具体实现

存储一个字符串，数据结构包含以下 3 种具体存储结构：

1. 定长顺序存储：实际上就是用普通数组（又称静态数组）存储。例如 C 语言使用普通数据储存字符串的代码为 `char a[20] = "data.biancheng.net"`；
2. 堆分配存储：用动态数组存储字符串；
3. 块链存储：用链表存储字符串；

以上 3 种存储结构会在后续文章中作详细介绍。

[联系方式](#) [购买教程（带答疑）](#)