

13 串 | 13A ADT

#数据结构邓神

定义

❖ 由来自字母表 Σ 的字符所组成的有限序列：

$$S = a_0 a_1 a_2 \dots a_{n-1} \in \Sigma^*$$

❖ 既然如此，为何不直接用序列来实现串？

❖ 通常，字符的种类不多，而串长 $= n \gg |\Sigma|$

❖ 比如：

英文文章	$(['A' - 'Z'] \cup ['a' - 'z'] \cup \{ ' ', '.', ',', '...', \dots \})^*$
C/C++程序	$(\{ 95 \text{个可打印字符} \} \cup \{ \text{LF}, \text{CR} \})^*$
天然蛋白质	$\{ 21 \text{种氨基酸} \}^*$
DNA	$\{ A, C, G, T \}^*$
RNA	$\{ A, C, G, U \}^*$
二进制	$\{ 0, 1 \}^*$

Linear

0 1 2 i n-1

Data Structures (Spring 2004), Tsinghua University

Bitmap?

我们是否可以压缩存储？

术语

术语

❖ 相等： $S[0, n) = T[0, m)$

长度相等 ($n = m$)，且对应的字符均相同 ($S[i] = T[i]$)

❖ 子串： $S.\text{substr}(i, k) = S[i, i + k)$, $0 \leq i < n$, $0 \leq k$

亦即，从 $S[i]$ 起的连续 k 个字符

❖ 前缀： $S.\text{prefix}(k) = S.\text{substr}(0, k) = S[0, k)$, $0 \leq k \leq n$

亦即， S 中最靠前的 k 个字符

❖ 后缀： $S.\text{suffix}(k) = S.\text{substr}(n - k, k) = S[n - k, n)$, $0 \leq k \leq n$

亦即， S 中最靠后的 k 个字符

❖ 联系： $S.\text{substr}(i, k) = S.\text{prefix}(i + k).\text{suffix}(k)$

❖ 空串： $S[0, n = 0)$ ，也是任何串的子串、前缀、后缀

ADT

length()	[0, n)	
charAt(i)	[0, i)	[i, (i, n)
substr(i, k)	[0, i)	[i, i + k) [i + k, n)
prefix(k)	[0, k) [k, n)	
suffix(k)	[0, n - k)	[n - k, n)
concat(T)	S	T
equal(T)	S	indexOf(P)
	T	[k, k + m) P[0, m)