

字符串哈希 & 字典树

QwQ

邵逸帆

23 电信基地班
兰州大学算法与程序设计集训队

2024 年 7 月 13 日



① 字符串基础

② 字符串哈希

③ 字典树

字符类型

- `char` 类型
- `char[]` 类型 (`char` 数组)
- `char*` 类型 (`char` 指针)

字符串字面量

eg. `"Hello, World!"`

- type: `char[14]` in C, `const char[14]` in C++
- size: 14

Warning: 不建议使用 `char* str = "Hello, World!"`; 这样的代码, 因为对于其的修改会导致未定义行为 (因为你在修改只读内存)。

我们可以使用字符数组来存储字符串, 这很直观。

`std::string` 与 `std::vector<char>` 非常相似，只不过前者提供了更多的字符串操作函数。当然，相较于 `char[]` 更有优势。

`std::string`

- `append()` & `operator+=`: 字符串拼接
- `find()`: 查找字符串中的某个子串
- `substr()`: 返回字符串的子串
- `operator==` & `<` & `>` etc.: 字符串比较

诸如 `std::to_string()`, `std::stoi()`, `isdigit()` 等函数有时候会帮助你更优雅地处理字符串。

字符串定义于字符集之上。一个建立了全序关系的集合 Σ 就可以作为一个字符集，其中的元素称为字符。字符串是字符的有限序列。

字符串的长度是指字符串中字符的个数。字符串的长度可以为 0。

之后的内容涉及到子串、子序列、前缀、后缀等概念，请注意区分。

① 字符串基础

② 字符串哈希

③ 字典树

pass

① 字符串基础

② 字符串哈希

③ 字典树

pass

THX 4 Listening! :)