

Matrix 大作业

助教：陆嘉馨、王一

本作业原始来源为程序设计2017的 OOP 大作业，本文档从原始manual中修改而成。

说明

这是程序设计B班和工科班的OOP大作业。请大家按照文档中的要求完成 `Matrix.cpp` 中的内容。

大家需要手动维护一位数据结构，以行优先的方式实现二维表示，**禁止使用vector、deque等任何STL容器**，需要通过正确性和鲁棒性测试。

接口

构造函数与赋值

- 默认构造函数
- `Matrix(size_t n, size_t m, T _init = T())`，构造一个大小为 $n \times m$ 的矩阵，并将里面每个元素初始化为 `init`；`Matrix(std::pair<size_t, size_t> sz, T _init = T())` 类似以上，其中 `sz.first = n, sz.second = m`。
- `Matrix(std::initializer_list<std::initializer_list<T>> il)` 利用 `initializer_list` 构造矩阵。`initializer_list` 的使用方法可以去参考资料中的网页搜索。
- “拷贝”构造与赋值。给定一个 `Matrix<U>` 构造 `Matrix<T>`。此部分保证给定的 `U` 可以构造 `T`。
- “移动”构造函数与赋值。仅要求相同类型的移动构造。

元素获取

完成如下两个函数（请务必写对），它们用于获取矩阵的第 i 行 j 列的元素，0-based.

- `const T &operator()(size_t i, size_t j) const`
- `T &operator()(size_t i, size_t j)`

并完成如下两个函数，返回矩阵的第 i 行 / 列。

- `Matrix<T> row(size_t i) const`
- `Matrix<T> column(size_t i) const`

运算

一元运算

- `-`, 对矩阵中的每个元素取负
- `--`
- `+=`, 注意要求需要支持 `Matrix<int> += Matrix<double>`
- `*=`, 注意这里是数乘
- `Matrix tran() const`，返回当前矩阵的转置矩阵，并且**不得改变当前矩阵**。

二元运算

- `==`, 比较两个矩阵是否相同（元素值相同）
- `!=`, 比较两个矩阵是否不同
- `+`, 定义为友元函数
- `-`, 定义为友元函数
- `*`, 数乘，定义为友元函数

- *, 矩阵乘法, 定义为友元函数

注意, 对于 `Matrix<U> a` 和 `Matrix<V> b`, 若 `U+V` 的类型为 `T`, 则 `a+b` 的返回值类型为 `Matrix<U>`, 提示可以使用 `decltype` 类型说明符。

迭代器

要求实现一个 `random_access` 的迭代器 (具体要求可以见源代码和测试代码), 以及如下函数。

- `iterator begin()`
- `iterator end()`
- `T& operator*() const`, 返回当前迭代器指向的数据
- `T* operator->() const`, 返回迭代器的指针。
- `std::pair<iterator, iterator> subMatrix(std::pair<size_t, size_t> l, std::pair<size_t, size_t> r)`. `M`是当前矩阵的一个子矩阵, `l`和`r`分别为它的左上角和右下角在原矩阵中的位置, 返回 `M` 的 `begin()` 和 `end()`。

其他

- `void clear()`, 清空当前矩阵, 并释放内存空间。
- `std::size_t rowlength()`, 返回行数。
- `std::size_t columnlength()`, 返回列数。
- `std::pair<std::size_t, std::size_t> size() const`, 返回矩阵大小。
- `void resize(std::size_t n, std::size_t m, T _init = T())`. 保留前 $n \times m$ 个元素, 若元素不足则拿 `_init` 补充, 并重新以行优先方式组成新矩阵。若元素个数相同, 则不允许重新开设内存空间。
- `void resize(std::pair<std::size_t, std::size_t> sz, T _init= T())`. 要求同上。

其他说明

- 整个项目基于C++14 标准完成。
- 编译命令: `g++ test.cpp -o test -std=C++14 -O2 -Wall -lopenblas`
- 尽可能项目中少出现Warning, 有助于减少不必要的bug。
- 请大家注意代码格式, 极为糟糕的代码风格会酌情扣分!
- Code review的时候会针对你写的代码和给定框架中的一些设计进行提问。
- 部分未在这里详细说明了的函数大家一般可以自行理解出其所要实现的内容, 对具体实现要求还有疑惑的也可以直接阅读给出的测试代码。

参考资料

大家有什么问题, 可以先查一下这些网站:

1. en.cppreference.com/w/
2. www.cplusplus.com
3. stackoverflow.com

另外关于迭代器, 可以参考候捷所著的《STL 源码剖析》。

关于 C++11/14 的内容, 可以参考Scott Meyers 所著的《Effective Modern C++》以及之前提到的网站。