

# IterableSquareMatrix

IterableSquareMatrix - интерфейс, позволяющий итерироваться по элементам квадратной матрицы. Интерфейс включает в себя следующие операции:

- `IterableSquareMatrix::RowIterator iterate_rows(size_t)` -- создаёт константный итератор для прохода по ненулевым элементам заданной строки
- `IterableSquareMatrix::ColumnIterator iterate_columns(size_t)` -- создаёт константный итератор для прохода по ненулевым элементам заданного столбца
- `size_t size()` -- возвращает размер матрицы

Поддерживаемые операции для итераторов:

(*p* – переменная типа `RowIterator` или `ColumnIterator`, *x* – переменная типа `double`)

Чтение	<code>x = *p</code>
Доступ	<code>p-&gt;column</code> (для <code>RowIterator</code> ) <code>p-&gt;row</code> (для <code>ColumnIterator</code> )
Изменение	<code>p++</code> <code>++p</code>
Сравнение	<code>p==q</code> <code>p!=q</code>

Проект содержит следующие файлы:

- 1) `IterableSquareMatrix.cpp`, `IterableSquareMatrix.hpp` -- реализация абстрактного класса `IterableSquareMatrix`.
- 2) `IterableSparseMatrix.cpp`, `IterableSparseMatrix.hpp` -- реализация класса `IterableSparseMatrix`, позволяющего итерироваться по ненулевым элементам разреженной матрицы. `SparseVector.cpp` `SparseVector.hpp` -- реализация разреженного вектора - вспомогательный класс для `IterableSparseMatrix`.
- 3) `IterableDenseMatrix.cpp`, `IterableDenseMatrix.hpp` -- реализация класса `IterableDenseMatrix`, позволяющего итерироваться по ненулевым элементам плотной матрицы. `DenseVector.cpp` `DenseVector.hpp` -- реализация плотного вектора -- вспомогательный класс для `IterableDenseMatrix`.
- 4) `RowIterator.cpp`, `ColumnIterator.hpp` -- реализации однонаправленных константных итераторов по строкам и столбцам матрицы соответственно.

- 5) Info.cpp, Info.hpp -- файлы, хранящие описание и реализацию методов вспомогательной структуры, необходимой для хранения информации в итераторах.
- 6) GraphAlgorithms.cpp, GraphAlgorithms.hpp -- Реализация алгоритмов на графах.

Реализованы две задачи на графах:

1. Поиск вершин с наибольшим количеством входящих и исходящих рёбер

set <unsigned> max\_edges (const IterableSquareMatrix \*); -- функция получает указатель на матрицу смежности графа, возвращает набор вершин с максимальным числом входящих и исходящих рёбер

2. Поиск максимальной клики в графе

set <set <unsigned>> max\_cliques(const IterableSquareMatrix \*); -- функция получает указатель на матрицу смежности графа, возвращает набор наборов вершин, каждый из которых представляет максимальную клику.

7\*) Файлы sparse1.cpp dense1.cpp sparse2.cpp dense2.cpp содержат одинаковые тесты при использовании разных типов матриц.

В файлах sparse1.cpp и dense2.cpp содержатся тесты, в которых типы матриц используются по назначению.

В файлах dense1.cpp и sparse2.cpp соответственно содержатся такие же тесты, но с неподходящим типом матриц.

Измерение и сравнение времени выполнения, например, sparse1.cpp и dense1.cpp, позволяет увидеть преимущество использования разреженной матрицы в подходящем случае.

(используйте make show для тестирования)