

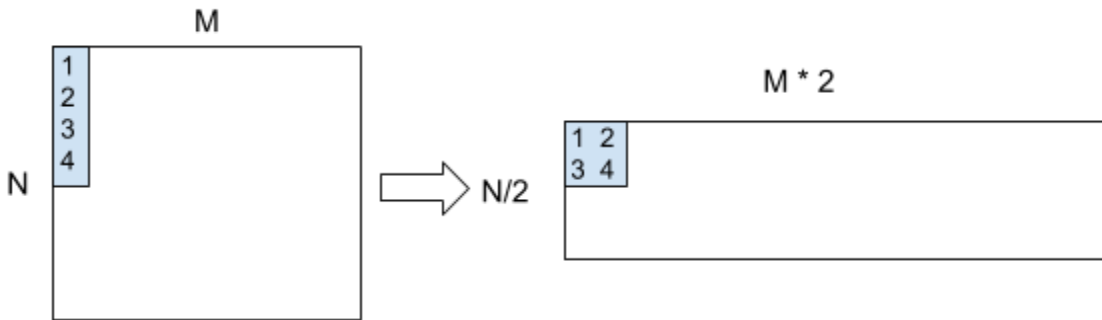
Общие требования:

- [методичка \(CUDA\) - черновик.pdf](#)
- реализация CPU и GPU
 - CPU - эталон (качество кода не существенно)
 - GPU - (каждая оптимизация - отдельное ядро, основанное на предыдущей реализации)
 - без транзакций в глобальную память - самый простой вариант
 - + работа с массивами, не кратными размеру блока и размерам обрабатываемого фрагмента
 - + корректный доступ в глобальную память
 - + разделяемая память и корректное обращение по банкам памяти
 - + применение pinned-памяти (продумать обоснование ее преимуществ)
 - + работа со сверхбольшими объемами данных (критерий - нехватка объема памяти GPU)
 - + применение технологии CUDA Stream
 - **1 задание = 1 ядро**
 - исключения:
 - при работе со сверхбольшими размерами матриц
- сравнение времени работы в едином формате
 - формат: микро-, миллисекунды и т.п.
 - измерение времени на GPU **через события** в CUDA
- сравнение результатов работы:
 - полное поэлементное сравнения массива
 - вывод фрагмента на экран
- тип данных:
 - char или short
 - **для претендования на бонусный балл**
 - int / float
 - для тех, у кого цель - сдать лаб.работу
 - доп.баллы не учитываются
- **допускается использование двух массивов: с входными и выходными данными. Дополнительные массивы запрещены**
- **инициализация случайными числами**
 - **ОПЦИЯ: инициализация массива через cuRAND**
- **Размер Grid > 1**
- **Размер Block > 32**
- **ОПЦИЯ: проверка ошибок выполнения cudaError_t**

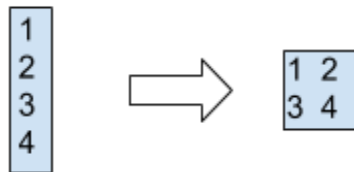
Балл от -0,5 до 0,5. Меняется линейно согласно следующим требованиям:

- списывание и непонимание кода - *-0,5 баллов*
- списывание и понимание кода - *0 баллов*
- работа с квадратной матрицей - *0 баллов*
- работа с матрицами/массивами кратными размеру grid и block - *0 баллов*
- работа с матрицами/массивами разного размера - *0,1 баллов*
- оптимизация кода - *до 0,2 балла*:
 - разделяемая память + корректное обращение по банкам памяти
 - формирование транзакций по 128 байт
 - обоснование, почему невозможно достигнуть 100% выполнения условия:
 - корректных транзакций в глобальную память
 - подсказка: добиться транзакции в глобальную память для всех заданий - !!!!! реально !!!!!
 - корректного обращения по банкам памяти
- работа со сверхбольшими объемами данных - *0,2 балла*
 - невозможно вычислить за один вызов ядра
 - объем матрицы / массива больше объема видеопамати

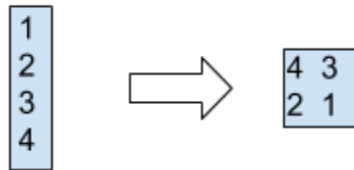
Вариант 1:



1. Входное окно 4x1. Выходное окно 2x2. Порядок значений прямой



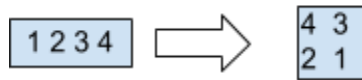
2. Входное окно 4x1. Выходное окно 2x2. Порядок значений обратный



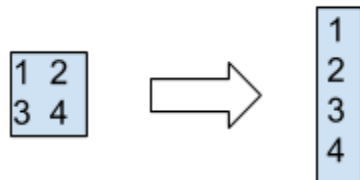
3. Входное окно 1x4. Выходное окно 2x2. Порядок значений прямой



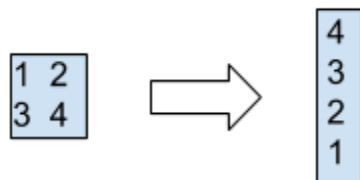
4. Входное окно 1x4. Выходное окно 2x2. Порядок значений обратный



5. Входное окно 2x2. Выходное окно 4x1. Порядок значений прямой



6. Входное окно 2x2. Выходное окно 4x1. Порядок значений обратный



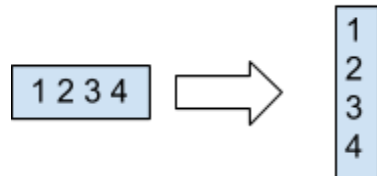
7. Входное окно 2x2. Выходное окно 1x4. Порядок значений прямой



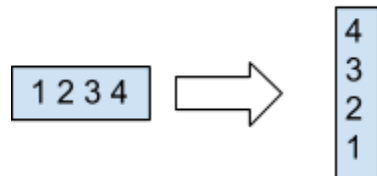
8. Входное окно 2x2. Выходное окно 1x4. Порядок значений обратный



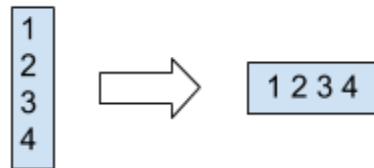
9. Входное окно 1x4. Выходное окно 4x1. Порядок значений прямой



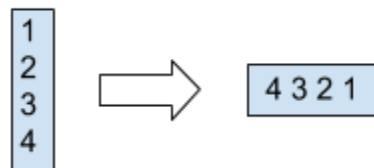
10. Входное окно 1x4. Выходное окно 4x1. Порядок значений обратный



11. Входное окно 4x1. Выходное окно 1x4. Порядок значений прямой

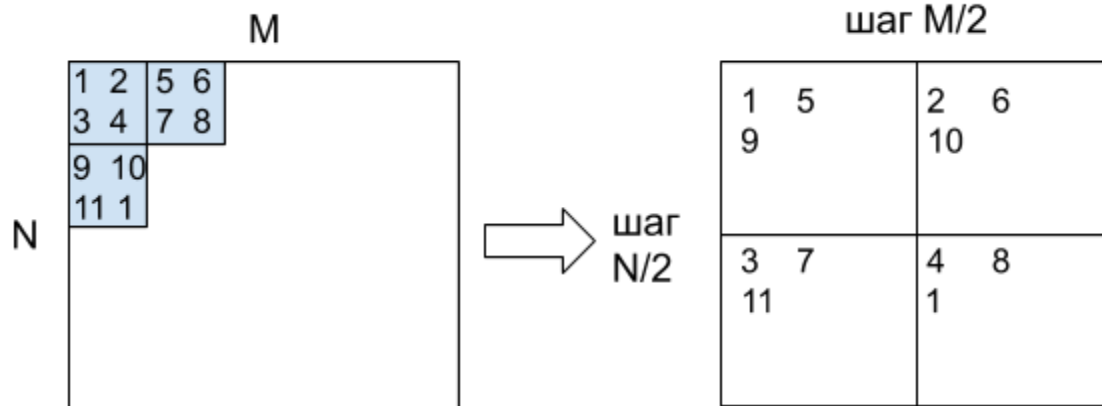


12. Входное окно 4x1. Выходное окно 1x4. Порядок значений обратный



Вариант 2:

В исходной матрице NxM, используя окно, переставить элементы с шагом N/2 и M/2. Размер выходной матрицы меняется пропорционально в зависимости от заданного окна.



1. входное окно 2×2

| | |
|---|---|
| 1 | 2 |
| 3 | 4 |

2. входное окно 1×4

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
|---|---|---|---|

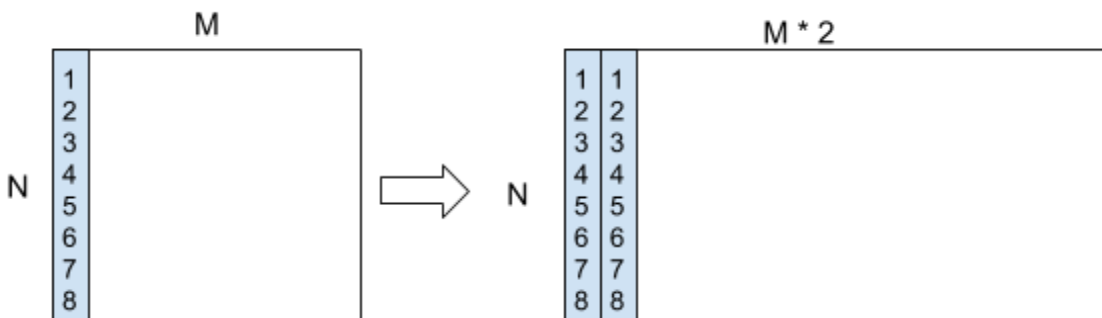
3. входное окно 4×1

| |
|---|
| 1 |
| 2 |
| 3 |
| 4 |

4. Другие значения шага

Вариант 3:

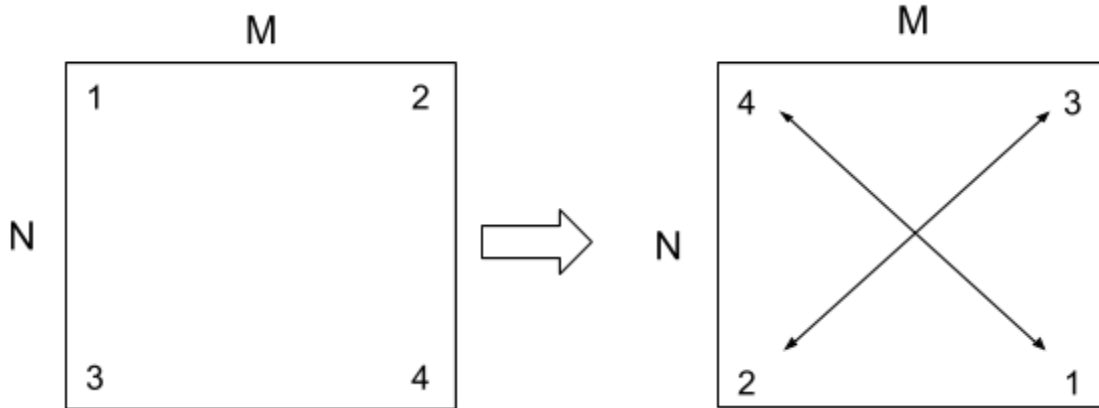
Из исходной матрицы $N \times M$ получить матрицу $N \times M \cdot 2$, продублировав столбцы



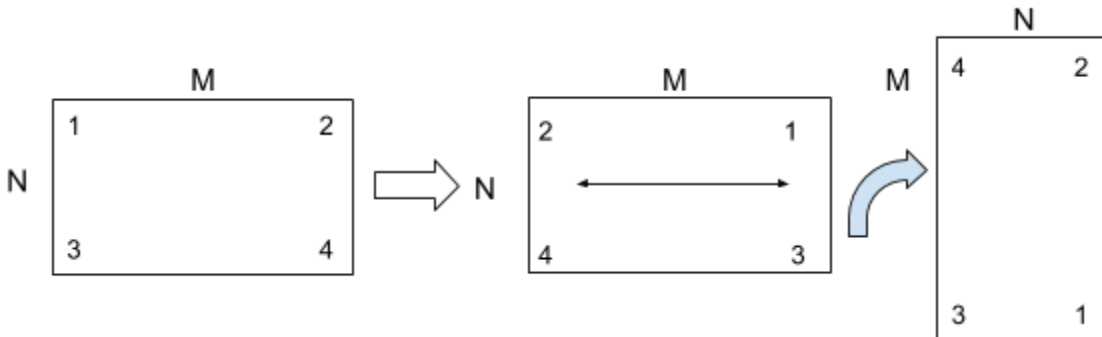
Вариант 4:

Отразить или повернуть матрицу

1. Отразить матрицу следующим способом



2. Отразить справа налево и повернуть по часовой стрелке на 90 гр.

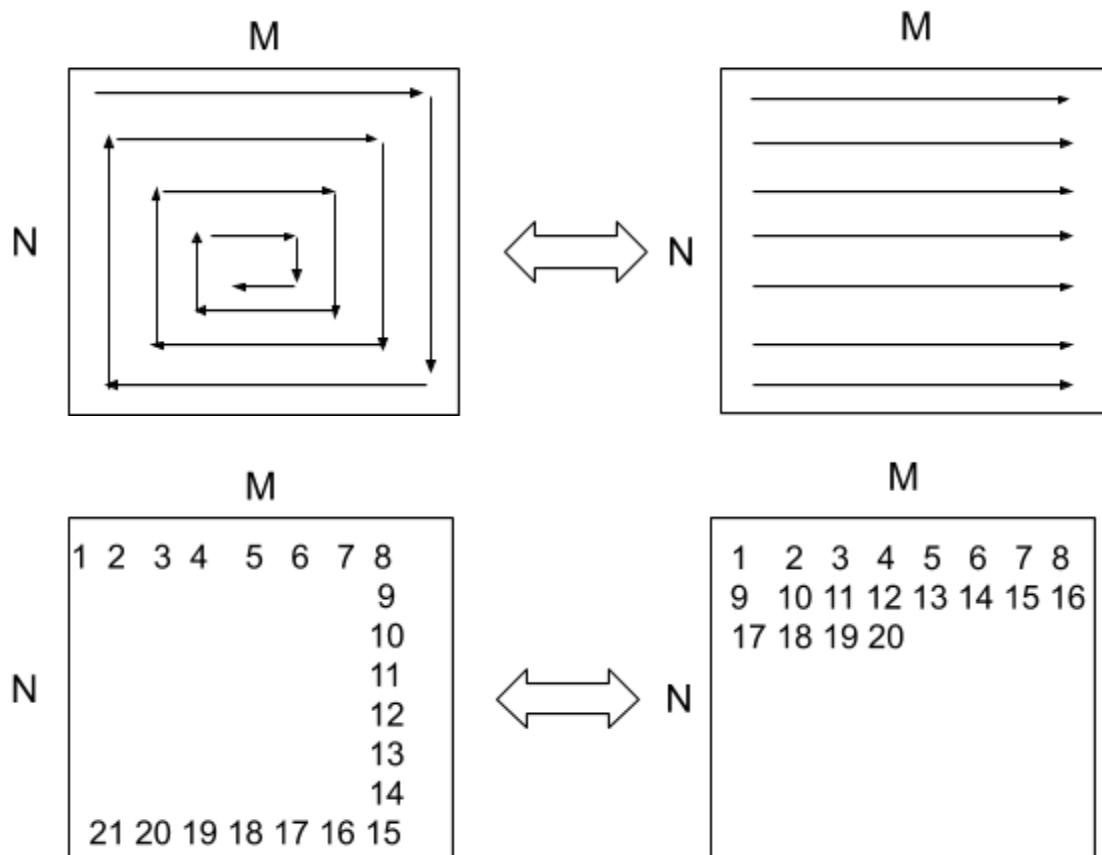


3. Другие варианты комбинаций

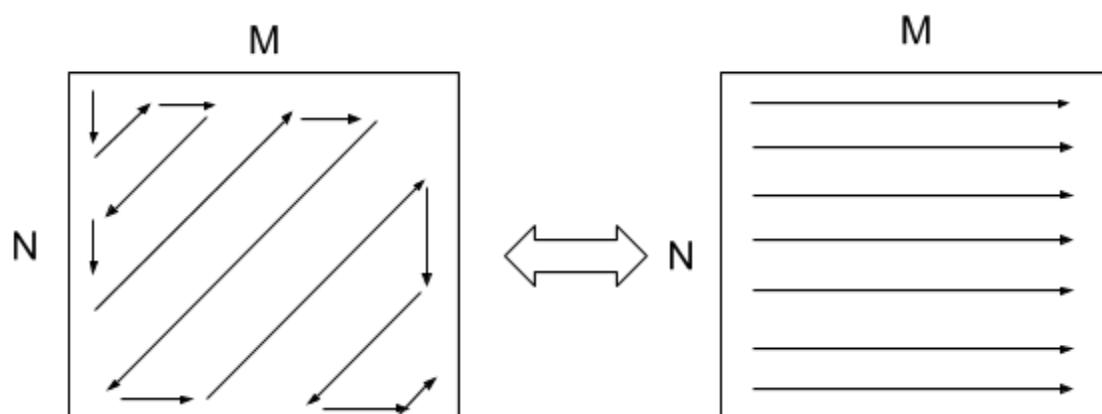
Вариант 5:

Из исходной матрицы получить выходную матрицу или массив, переставив элементы следующий образом. Возможно обратное преобразование. Элементы проинициализированы случайным образом.

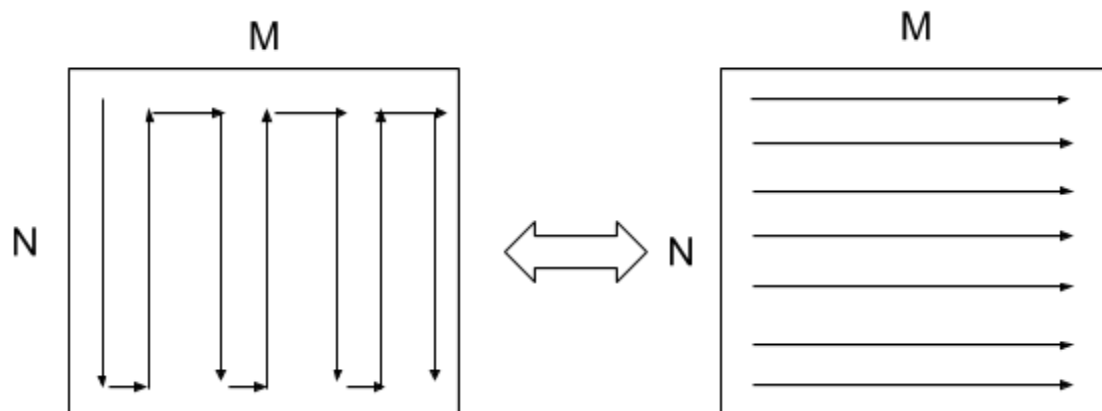
1. Порядок обхода исходной матрицы по спирали (по часовой или против часовой стрелки)



2. Порядок обхода исходной матрицы по диагонали



3. Порядок обхода исходной матрицы змейкой сверху вниз (справа налево)



4. Другие варианты обхода