

# Вычисления на видеокартах



## Лекция 8 - Разреженные матрицы

- COO, CSR, **CSC**
- ELL, DIA, HYB
- CHM, Disjoint Set

(Системы Непересекающихся Множеств)

 [@UnicornGlade](https://t.me/UnicornGlade)  
 [@PolarNick239](https://t.me/PolarNick239)  
 [polarnick239@gmail.com](mailto:polarnick239@gmail.com)

 Николай Полярный

Activate Ubuntu  
Go to Settings to activate Ubuntu<sup>1</sup>.

# Глава 1: Разреженные матрицы





**Task05: Radix sort на Tesla T4 (320 GB/s)**



[Роберт Смайт](#) - команда СПбГУ - **1553 uint millions/s**, CUDA



[Andrew Ratkov](#) - команда СПбГУ - **451 uint millions/s**, OpenCL



[Pribytkov Fedor](#) - команда СПбГУ - **257 uint millions/s**, OpenCL

P.S. мое baseline решение - **447 millions/s**, CPU std::sort - **9 millions/s**

# Представления разреженных матриц

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Что если число ненулевых значений **NNZ=11**  
много меньше  **$N \cdot M$** ? Например  **$NNZ = O(N)$**

**Какие с этим проблемы?**

# Представления разреженных матриц

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 | 0 | 11 | 0  |
| 1 | 0 | 3 | 4 | 0  | 0  |
| 2 | 0 | 5 | 6 | 7  | 0  |
| 3 | 0 | 0 | 0 | 8  | 0  |
| 4 | 0 | 0 | 0 | 9  | 10 |

Что если число ненулевых значений **NNZ=11**  
много меньше  **$N \cdot M$** ? Например  **$NNZ = O(N)$**

Какие с этим проблемы?

- слишком много **VRAM** зря
- слишком много **Bandwidth** в холостую
- слишком много **GFLOPS** в холостую

# Представления разреженных матриц

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 | 0 | 11 | 0  |
| 1 | 0 | 3 | 4 | 0  | 0  |
| 2 | 0 | 5 | 6 | 7  | 0  |
| 3 | 0 | 0 | 0 | 8  | 0  |
| 4 | 0 | 0 | 0 | 9  | 10 |

Что если число ненулевых значений **NNZ=11**  
много меньше **N\*M**? Например **NNZ=O(N)**

Какие с этим проблемы?

- слишком много **VRAM** зря
- слишком много **Bandwidth** в холостую
- слишком много **GFLOPS** в холостую



Это наше вдохновение!  
Это наша точка роста!  
Это нас **МАНИТ**!

# Представления разреженных матриц

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 | 0 | 11 | 0  |
| 1 | 0 | 3 | 4 | 0  | 0  |
| 2 | 0 | 5 | 6 | 7  | 0  |
| 3 | 0 | 0 | 0 | 8  | 0  |
| 4 | 0 | 0 | 0 | 9  | 10 |

Что если число ненулевых значений **NNZ=11** много меньше  **$N \cdot M$** ? Например  **$NNZ = O(N)$**

Какие с этим проблемы?

- слишком много **VRAM** зря
- слишком много **Bandwidth** в холостую
- слишком много **GFLOPS** в холостую

Как представить разреженные матрицы?

Какие требования предъявим к форматам?

# Представления разреженных матриц

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 | 0 | 11 | 0  |
| 1 | 0 | 3 | 4 | 0  | 0  |
| 2 | 0 | 5 | 6 | 7  | 0  |
| 3 | 0 | 0 | 0 | 8  | 0  |
| 4 | 0 | 0 | 0 | 9  | 10 |

Что если число ненулевых значений **NNZ=11** много меньше  **$N \cdot M$** ? Например  **$NNZ = O(N)$**

**Какие с этим проблемы?**

- слишком много **VRAM** зря
- слишком много **Bandwidth** в холостую
- слишком много **GFLOPS** в холостую

**Хотим:**

- компактность (пропорционально **NNZ**)
- быстрое создание
- быстрое умножение

# 1) COOrdinate-wise - COO

Что **САМОЕ** простое можно придумать?

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

# 1) COOOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|---|
| row: |   |   |   |   |   |   |   |   |   |   |
| col: |   |   |   |   |   |   |   |   |   |   |
| val: | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |

**А в каком порядке?**

# 1) COOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

row:

?

col:

?

val:

1

2

1

1

3

4

5

6

7

8

9

10

# 1) COOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

row:

0

?

col:

0

?

val:

1

2

1

3

4

5

6

7

8

9

10

1

# 1) COOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

row:

|   |   |   |  |  |  |  |  |  |  |  |
|---|---|---|--|--|--|--|--|--|--|--|
| 0 | 0 | ? |  |  |  |  |  |  |  |  |
|---|---|---|--|--|--|--|--|--|--|--|

col:

|   |   |   |  |  |  |  |  |  |  |  |
|---|---|---|--|--|--|--|--|--|--|--|
| 0 | 1 | ? |  |  |  |  |  |  |  |  |
|---|---|---|--|--|--|--|--|--|--|--|

val:

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|

1

# 1) COOOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |   |   |   |   |   |    |  |  |
|------|---|---|---|---|---|---|---|---|---|---|----|--|--|
| row: | 0 | 0 | 0 | ? |   |   |   |   |   |   |    |  |  |
| col: | 0 | 1 | 3 | ? |   |   |   |   |   |   |    |  |  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  |  |

# 1) COOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |   |   |   |   |   |    |  |  |  |
|------|---|---|---|---|---|---|---|---|---|---|----|--|--|--|
| row: | 0 | 0 | 0 | 1 | ? |   |   |   |   |   |    |  |  |  |
| col: | 0 | 1 | 3 | 1 | ? |   |   |   |   |   |    |  |  |  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  |  |  |

# 1) COOOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |     |   |   |   |   |    |  |  |  |
|------|---|---|---|---|---|-----|---|---|---|---|----|--|--|--|
| row: | 0 | 0 | 0 | 1 | 1 | ... |   |   |   |   |    |  |  |  |
| col: | 0 | 1 | 3 | 1 | 2 | ... |   |   |   |   |    |  |  |  |
| val: | 1 | 2 | 1 | 3 | 4 | 5   | 6 | 7 | 8 | 9 | 10 |  |  |  |

# 1) COOOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |   |   |   |   |   |    |
|------|---|---|---|---|---|---|---|---|---|---|----|
| row: | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4  |
| col: | 0 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

# 1) COOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |   |   |   |   |   |    |
|------|---|---|---|---|---|---|---|---|---|---|----|
| row: | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4  |
| col: | 0 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Насколько все хорошо?

# 1) COOrdinate-wise - COO

|   | 0 | 1 | 2 | 3  | 4  |
|---|---|---|---|----|----|
| 0 | 1 | 2 |   | 11 |    |
| 1 |   | 3 | 4 |    |    |
| 2 |   | 5 | 6 | 7  |    |
| 3 |   |   |   | 8  |    |
| 4 |   |   |   | 9  | 10 |

Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |   |   |   |   |   |    |
|------|---|---|---|---|---|---|---|---|---|---|----|
| row: | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4  |
| col: | 0 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**Насколько все хорошо?**

**- компактность?**

# 1) COOrdinate-wise - COO

Список из (row, col, val)



???



Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |   |   |   |   |   |    |
|------|---|---|---|---|---|---|---|---|---|---|----|
| row: | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4  |
| col: | 0 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**Насколько все хорошо?**

- компактность (**NNZ + NNZ + NNZ**)

- как реализовать быстрое создание?

# 1) COOOrdinate-wise - COO

Список из (row, col, val)



Сортируем по  
key=(row, col)



Сохраним перечень **ненулевых** элементов:

|      |   |   |   |   |   |   |   |   |   |   |    |
|------|---|---|---|---|---|---|---|---|---|---|----|
| row: | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4  |
| col: | 0 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

**Насколько все хорошо?**

- КОМПАКТНОСТЬ (**NNZ + NNZ + NNZ**)

- **как реализовать быстрое создание?**

# 1) COOOrdinate-wise - COO

|      |   |   |   |   |   |   |   |   |   |   |    |
|------|---|---|---|---|---|---|---|---|---|---|----|
| row: | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4  |
| col: | 0 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Diagram illustrating the COO (Coordinate-wise) format for a sparse matrix. The matrix is represented by three arrays: row, col, and val. The row array contains indices of non-zero elements, the col array contains column indices, and the val array contains the values. Blue vertical lines separate the columns, and blue horizontal lines separate the rows. A blue triangle is shown above the row array, indicating the cumulative count of non-zero elements per column.

**Насколько все хорошо?**

- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- **что еще нам нужно?**

# 1) COOOrdinate-wise - COO

|      |   |   |   |   |   |   |   |   |   |   |    |
|------|---|---|---|---|---|---|---|---|---|---|----|
| row: | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4  |
| col: | 0 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Разреженная **СЛАУ**:  $A \cdot x = b$

Итеративные методы:

- Якоби
- Гаусса-Зейделя
- Сопряженных Градиентов
- Метод Рундсона:

$$x^{k+1} = x^k - \tau(Ax^k - b)$$

**Насколько все хорошо?**

- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание

**- что еще нам нужно?**

# 1) COOOrdinate-wise - COO

|      |   |   |   |   |   |   |   |   |   |   |    |
|------|---|---|---|---|---|---|---|---|---|---|----|
| row: | 0 | 0 | 0 | 1 | 1 | 2 | 2 | 2 | 3 | 4 | 4  |
| col: | 0 | 1 | 3 | 1 | 2 | 1 | 2 | 3 | 3 | 3 | 4  |
| val: | 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

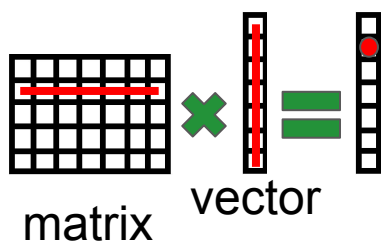
Разреженная **СЛАУ**:  $A \cdot x = b$

Итеративные методы:

- Якоби
- Гаусса-Зейделя
- Сопряженных Градиентов
- Метод Рундсона:

$$x^{k+1} = x^k - \tau(Ax^k - b)$$

**SpMV** (Sparse matrix–vector multiplication)



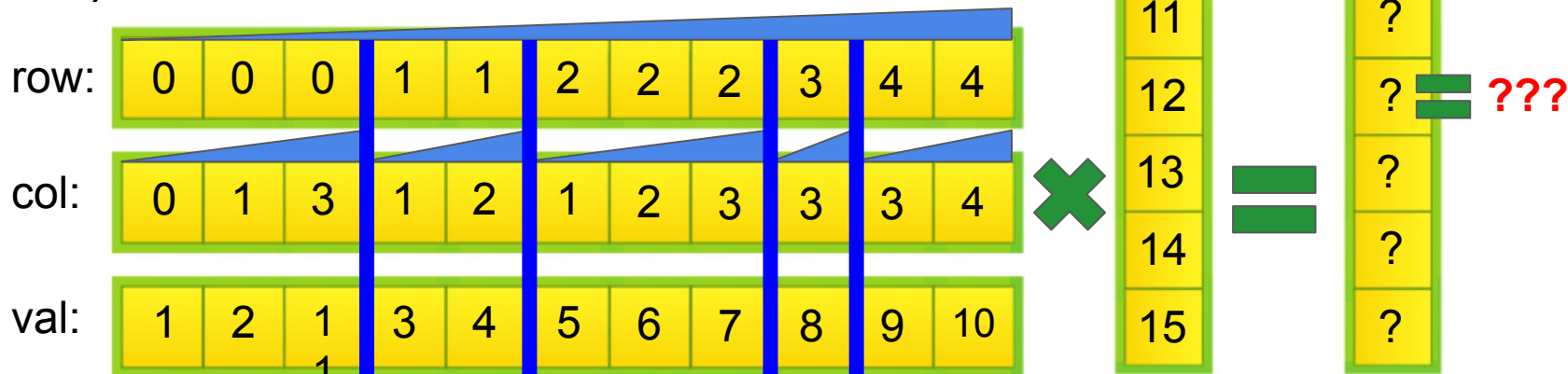
**Насколько все хорошо?**

- компактность (**NNZ + NNZ + NNZ**)

- быстрое создание

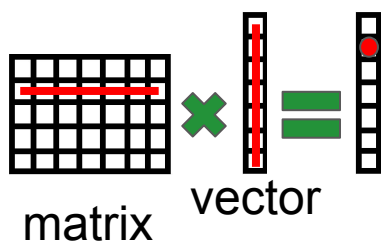
**- быстрое умножение?**

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

**SpMV** (Sparse matrix–vector multiplication)



**Насколько все хорошо?**

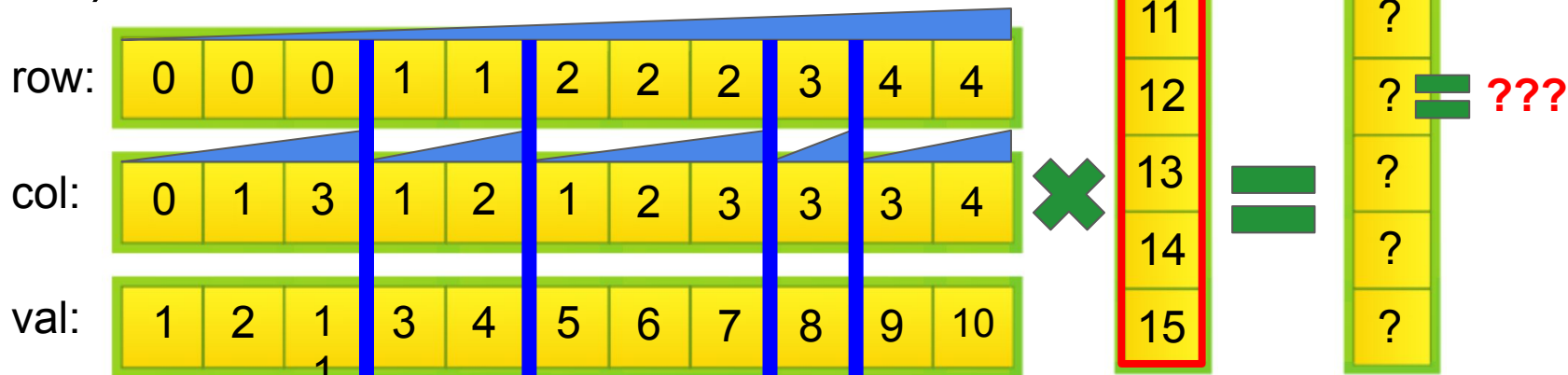
- компактность (**NNZ + NNZ + NNZ**)

- быстрое создание

**- быстрое умножение?**

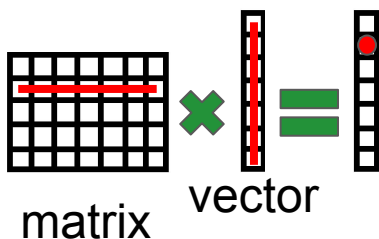
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

**SpMV** (Sparse matrix–vector multiplication)

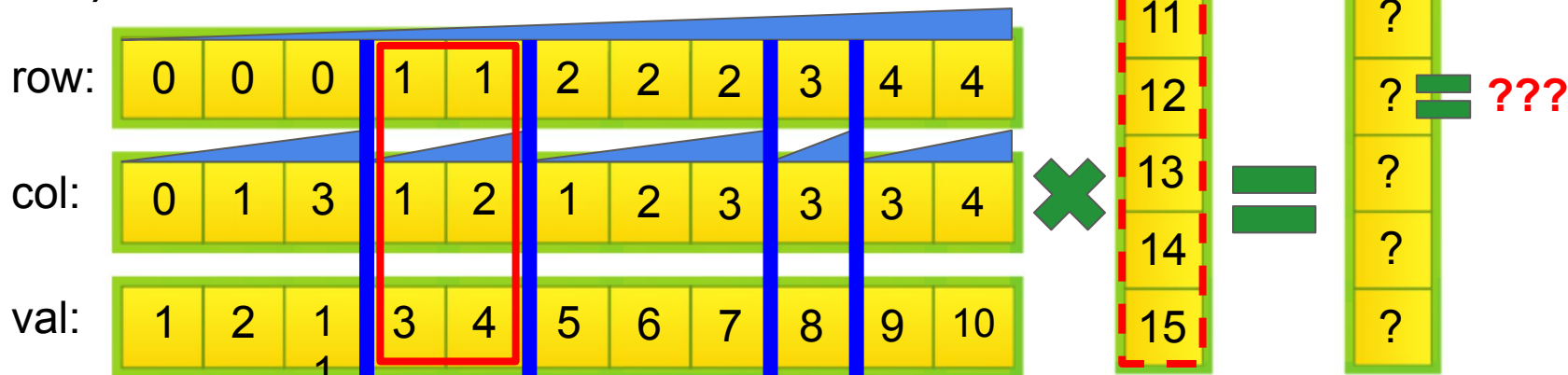


**Насколько все хорошо?**

- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- **быстрое умножение?**

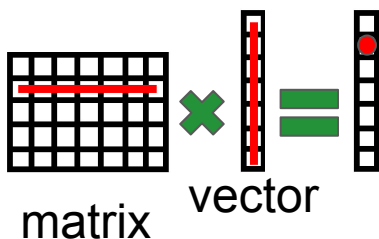
# 1) COOrdinate-wise - COO

Умножаем на  
какие элементы?



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

SpMV (Sparse matrix–vector multiplication)

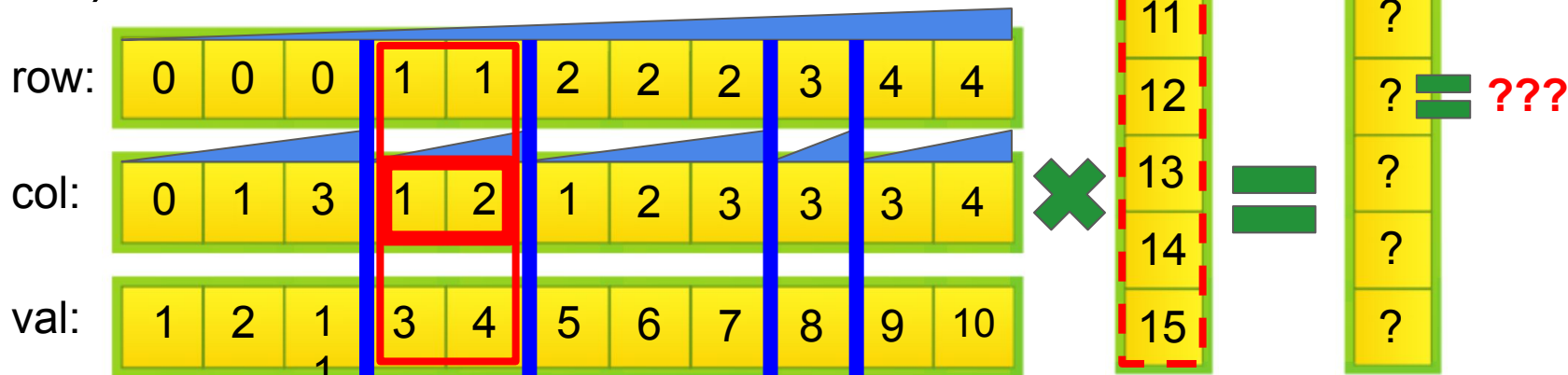


Насколько все хорошо?

- компактность ( $NNZ + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение?

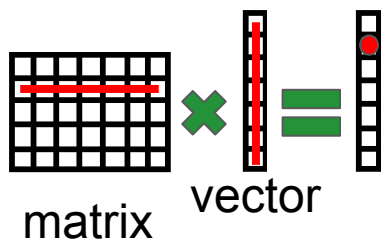
# 1) COOrdinate-wise - COO

Умножаем на  
какие элементы?



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

SpMV (Sparse matrix–vector multiplication)

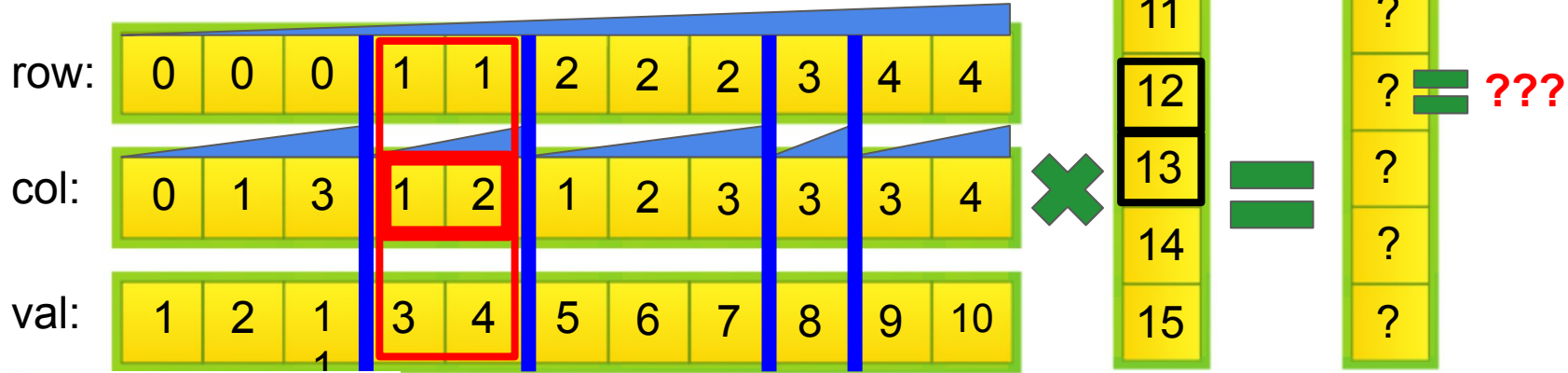


Насколько все хорошо?

- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- **быстрое умножение?**

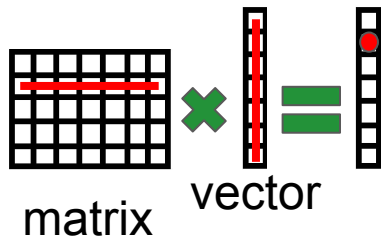
Умножаем на  
какие элементы?

# 1) COOOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

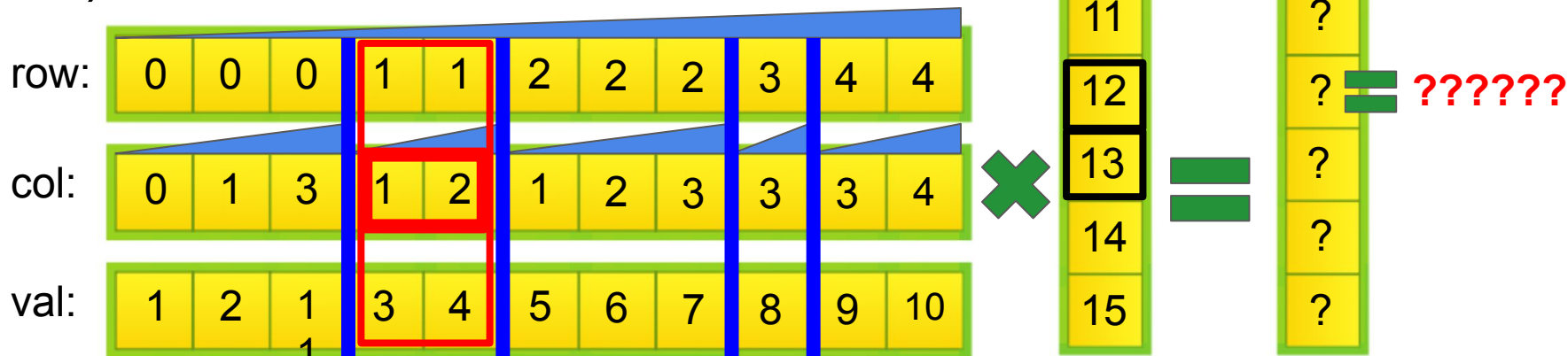
SpMV (Sparse matrix–vector multiplication)



Насколько все хорошо?

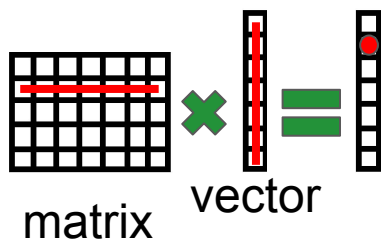
- компактность ( $NNZ + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение?

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

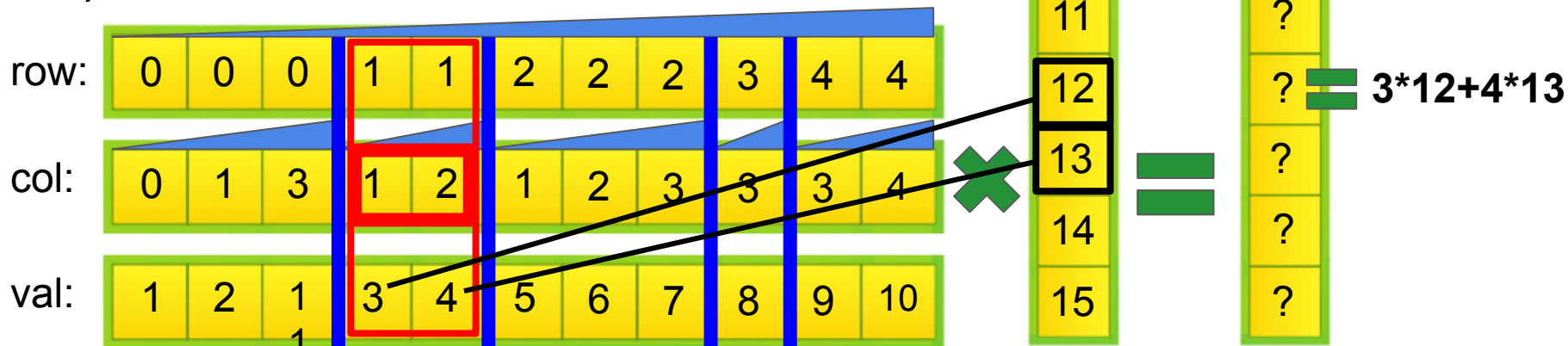
**SpMV** (Sparse matrix–vector multiplication)



**Насколько все хорошо?**

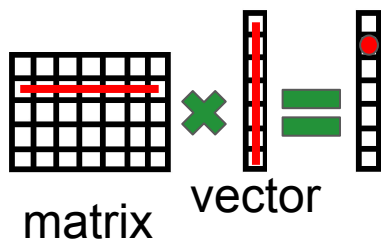
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- **быстрое умножение?**

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

**SpMV** (Sparse matrix–vector multiplication)



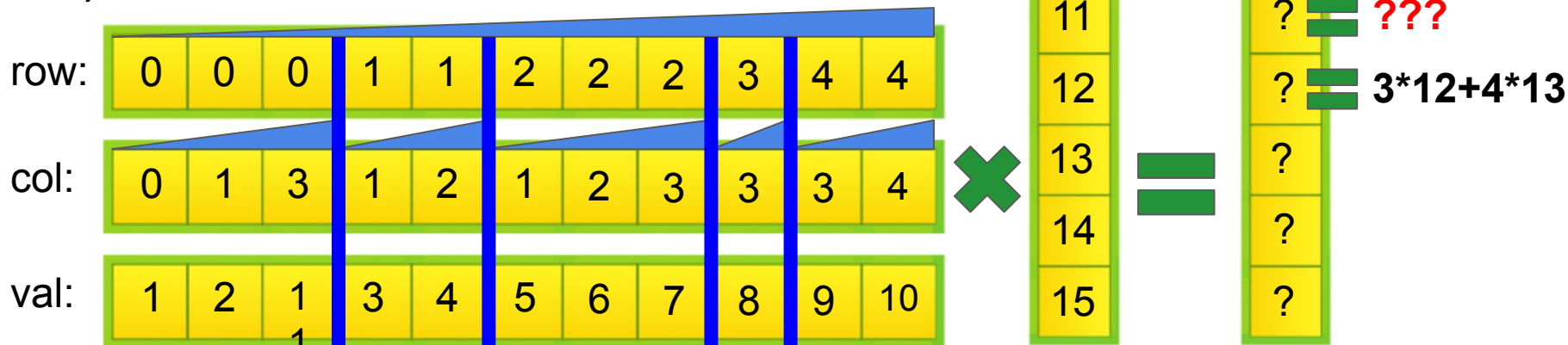
**Насколько все хорошо?**

- компактность (**NNZ + NNZ + NNZ**)

- быстрое создание

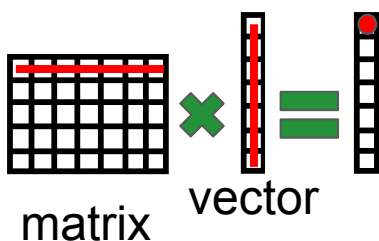
- **быстрое умножение?**

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

**SpMV** (Sparse matrix–vector multiplication)



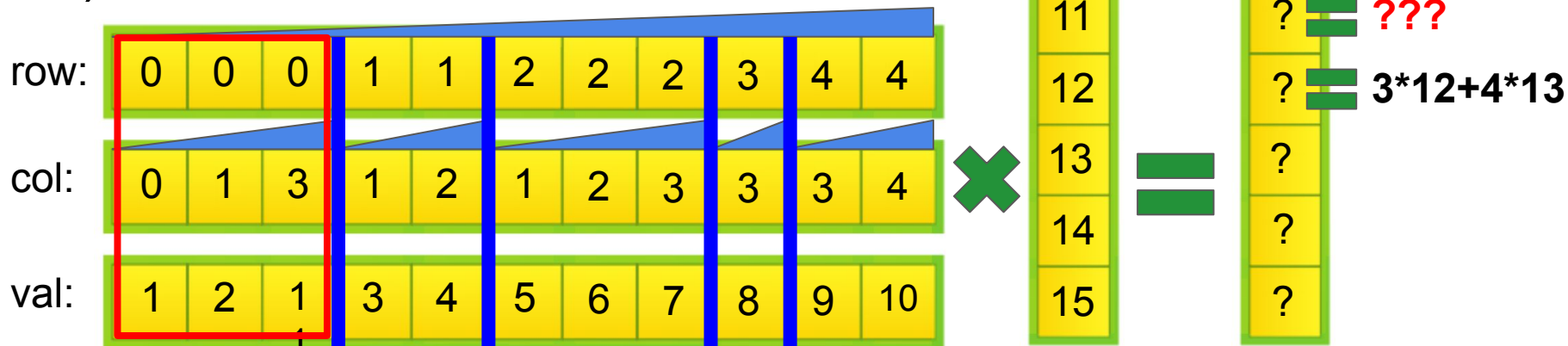
• **Насколько все хорошо?**

- компактность (**NNZ + NNZ + NNZ**)

- быстрое создание

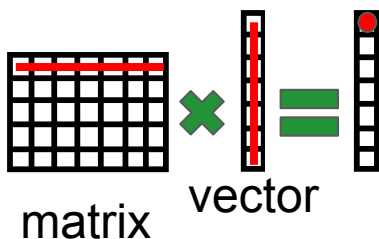
- **быстрое умножение?**

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

**SpMV** (Sparse matrix–vector multiplication)



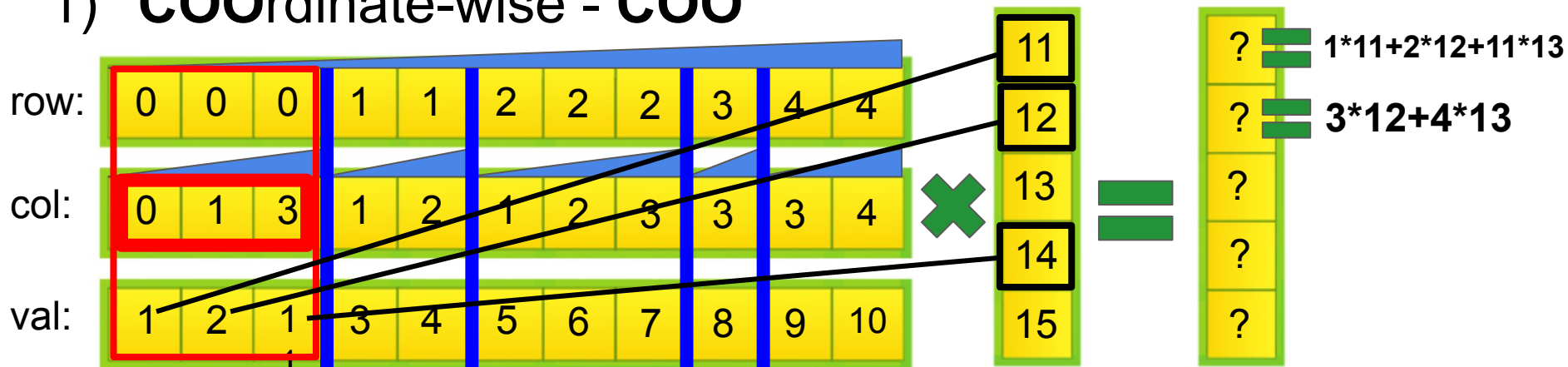
• **Насколько все хорошо?**

- компактность (**NNZ + NNZ + NNZ**)

- быстрое создание

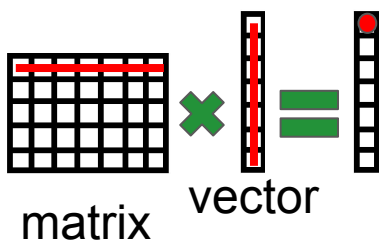
- **быстрое умножение?**

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

**SpMV** (Sparse matrix–vector multiplication)



• **Насколько все хорошо?**

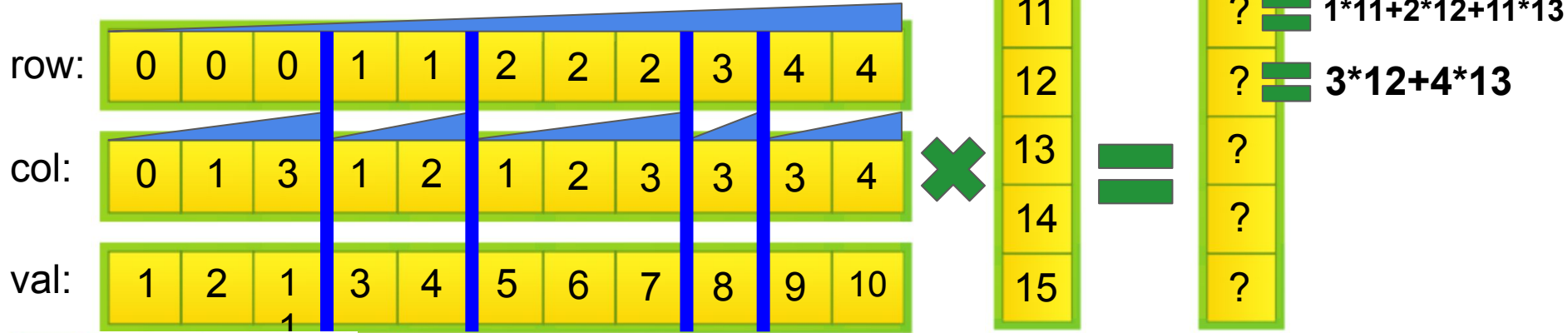
- компактность (**NNZ + NNZ + NNZ**)

- быстрое создание

- **быстрое умножение?**

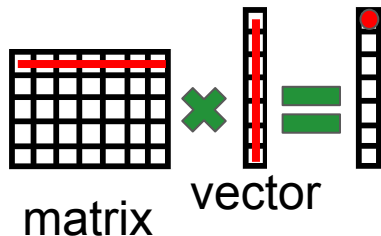
Что делает один WorkItem?

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

SpMV (Sparse matrix–vector multiplication)



• Насколько все хорошо?

- компактность ( $NNZ + NNZ + NNZ$ )

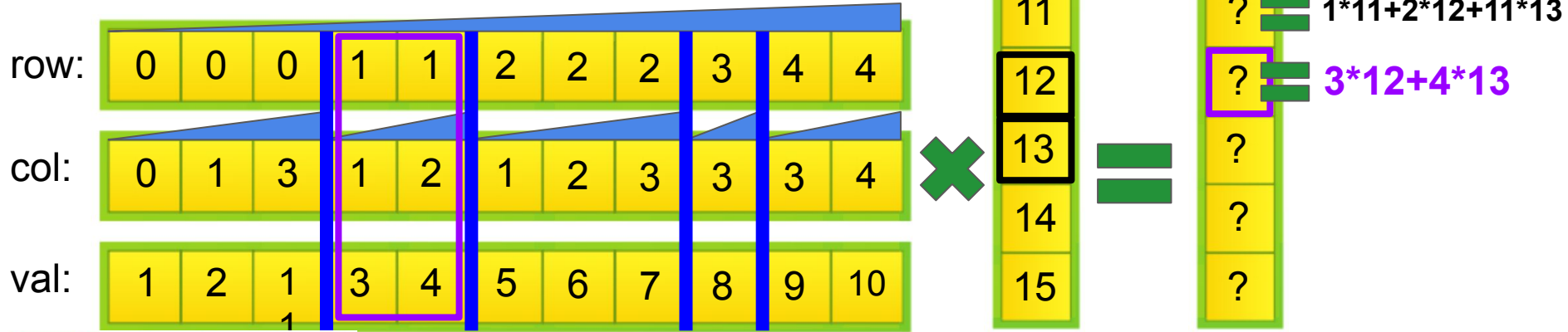
- быстрое создание

- быстрое умножение?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

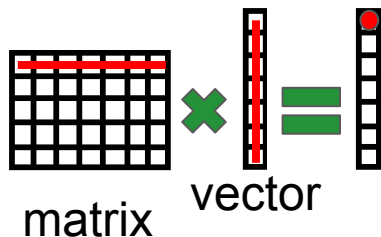
Что делает один WorkItem?

# 1) COOOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

SpMV (Sparse matrix–vector multiplication)



Насколько все хорошо?

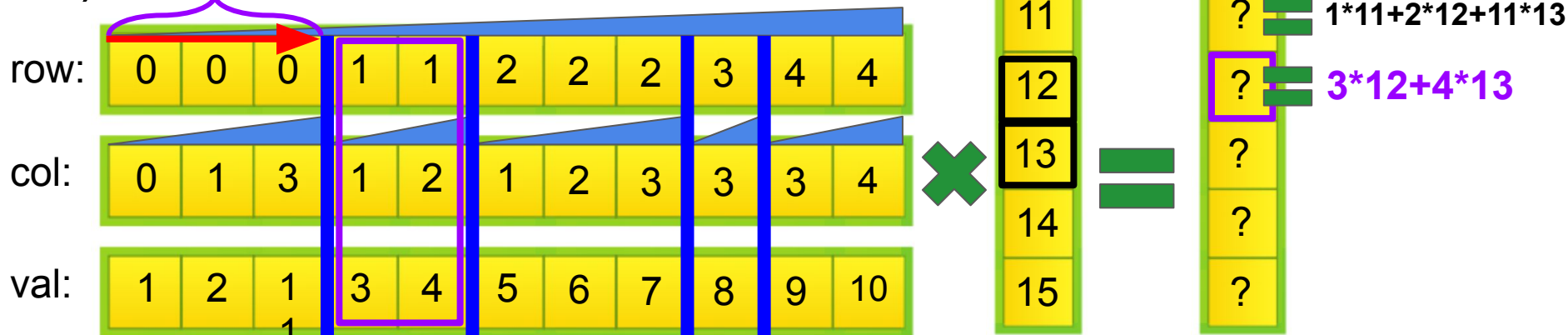
- компактность ( $NNZ + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

Как найти offset?

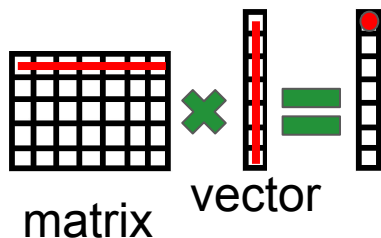
Что делает один WorkItem?

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

SpMV (Sparse matrix–vector multiplication)



Насколько все хорошо?

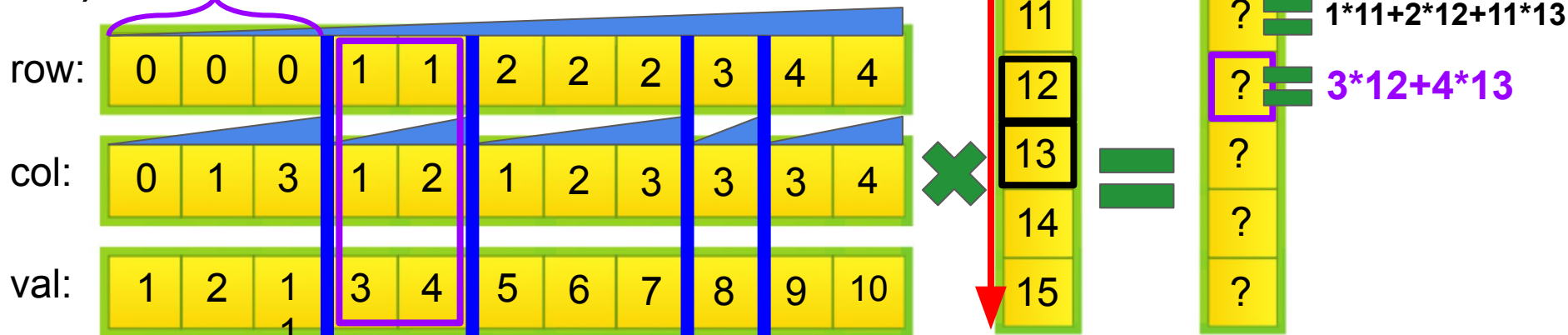
- компактность ( $NNZ + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

Как найти нужные элементы вектора?

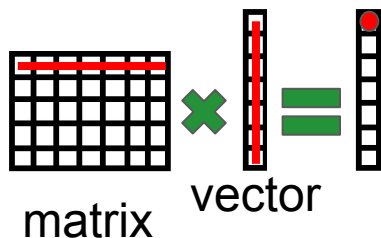
Что делает один WorkItem?

## 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

SpMV (Sparse matrix–vector multiplication)



Насколько все хорошо?

- компактность ( $NNZ + NNZ + NNZ$ )

- быстрое создание

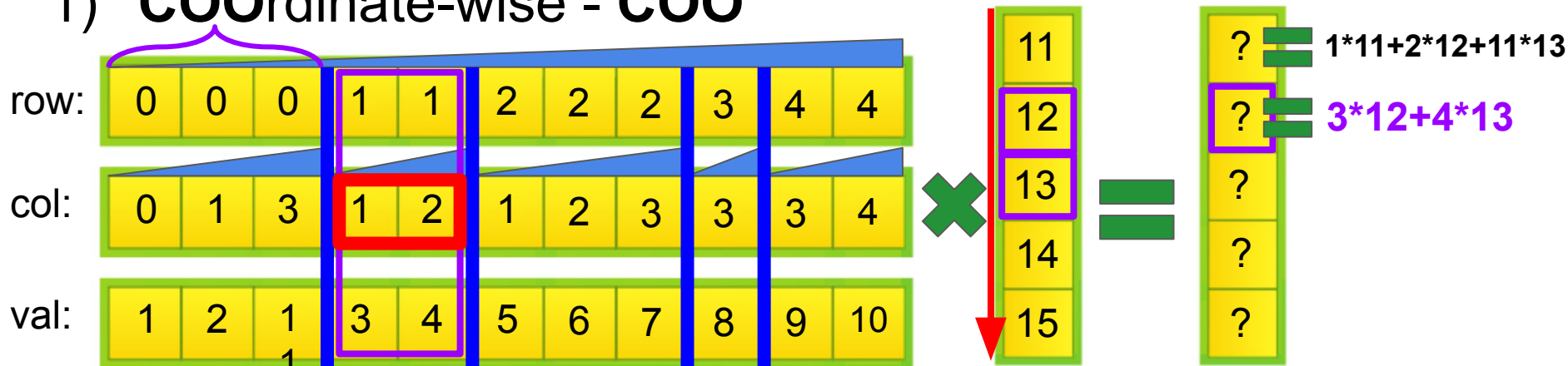
- быстрое умножение?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

Как найти нужные элементы вектора?

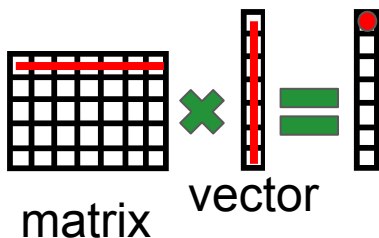
Что делает один WorkItem?

## 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

SpMV (Sparse matrix–vector multiplication)



Насколько все хорошо?

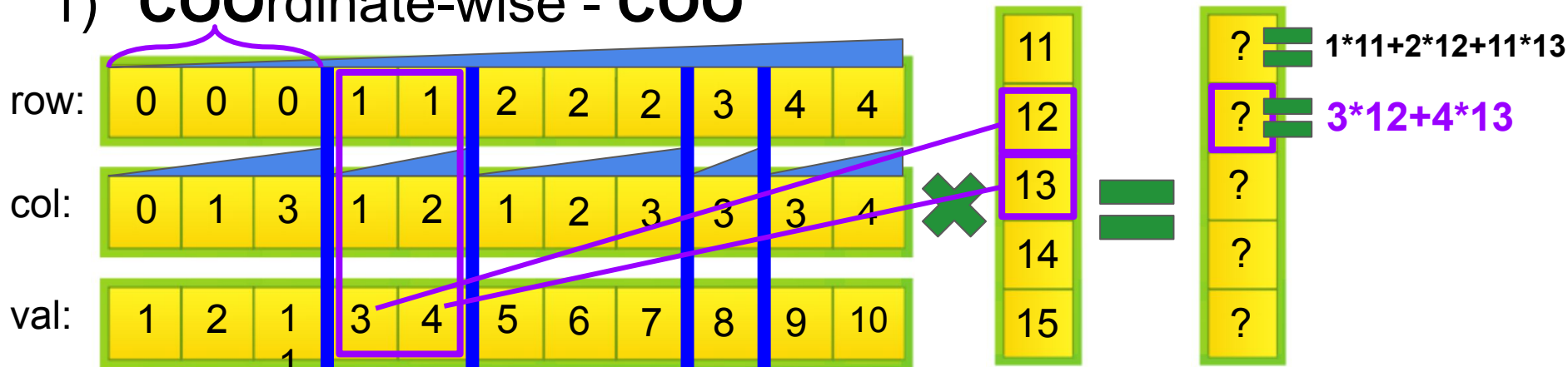
- компактность ( $NNZ + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

Как найти нужные элементы вектора?

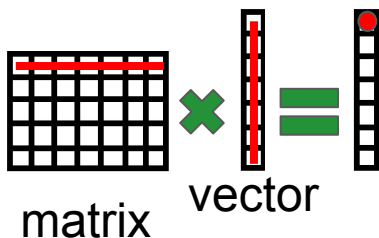
Что делает один WorkItem?

## 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

**SpMV** (Sparse matrix–vector multiplication)



Насколько все хорошо?

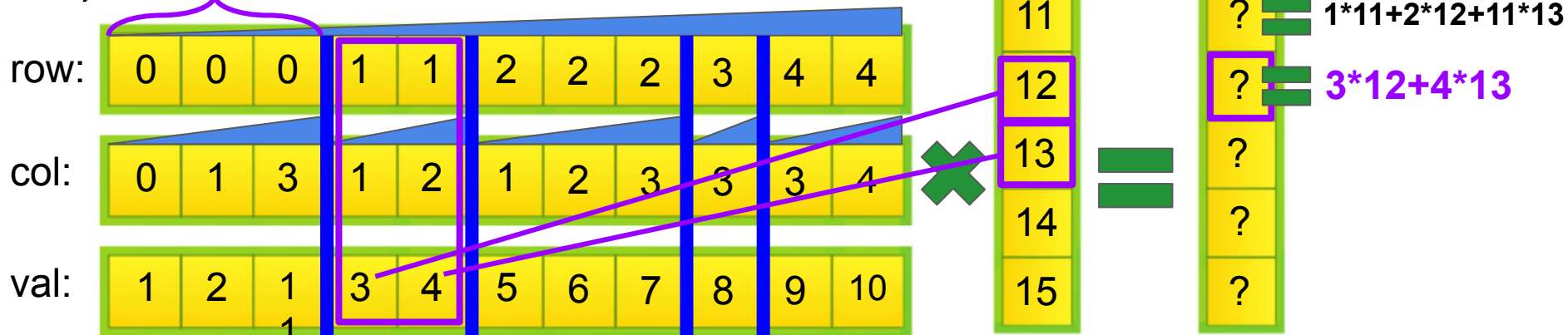
- компактность ( $NNZ + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

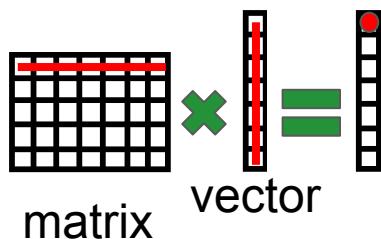
Что является узким местом?

Что делает один WorkItem?

# 1) COOrdinate-wise - COO



SpMV (Sparse matrix-vector multiplication)



Насколько все хорошо?

- компактность ( $NNZ + NNZ + NNZ$ )

- быстрое создание

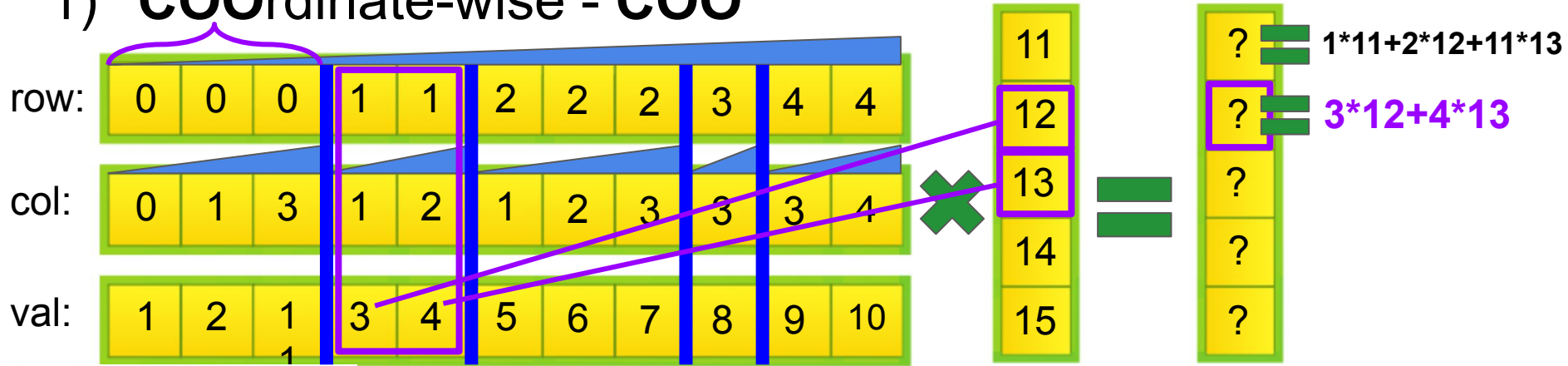
- быстрое умножение?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

Что является узким местом?  
*Coalesced* ли паттерн чтения?

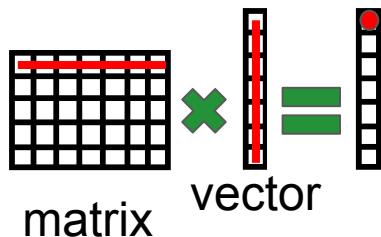
Что делает один **WorkItem**?

# 1) COOrdinate-wise - COO



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

**SpMV** (Sparse matrix–vector multiplication)



Насколько все хорошо?

- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- **быстрое умножение?**

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

## 2) CSR - Compressed Sparse Row

row

offset: ???

col:



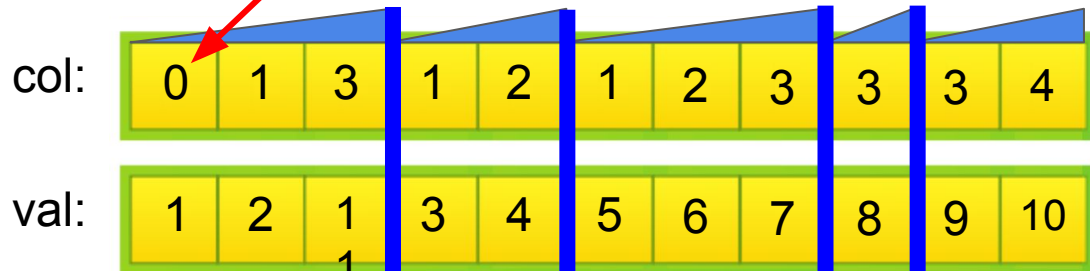
val:



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

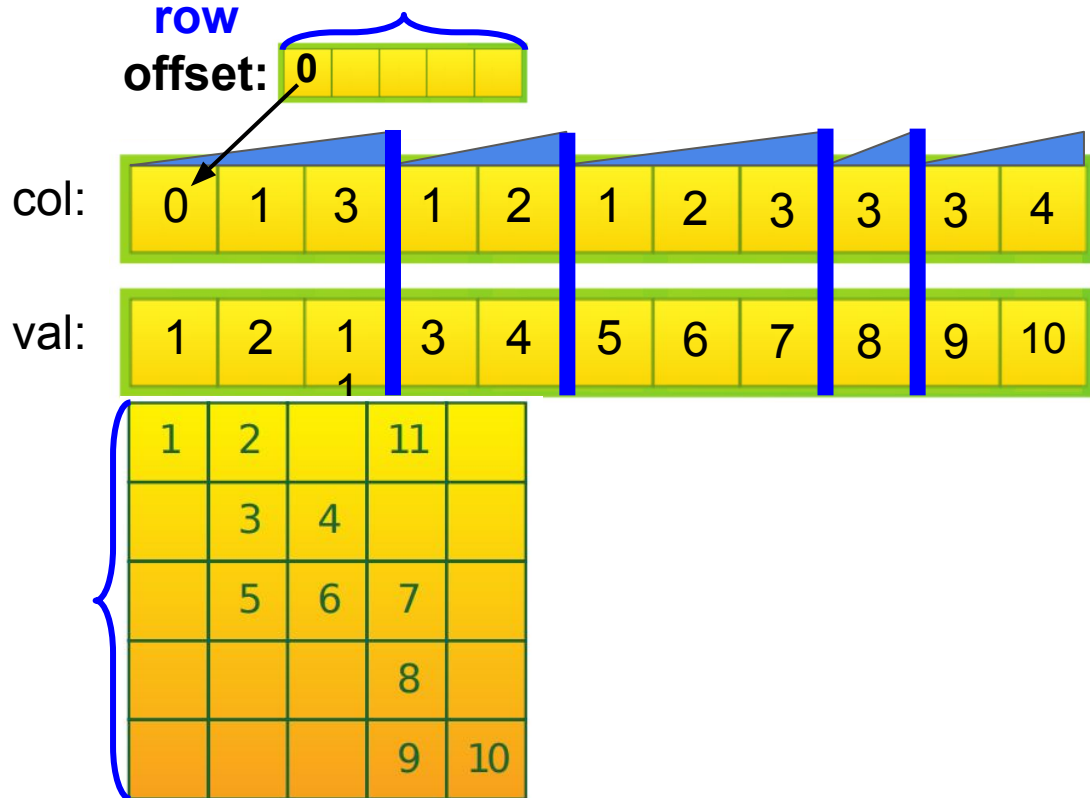
## 2) CSR - Compressed Sparse Row

row  
offset: ?

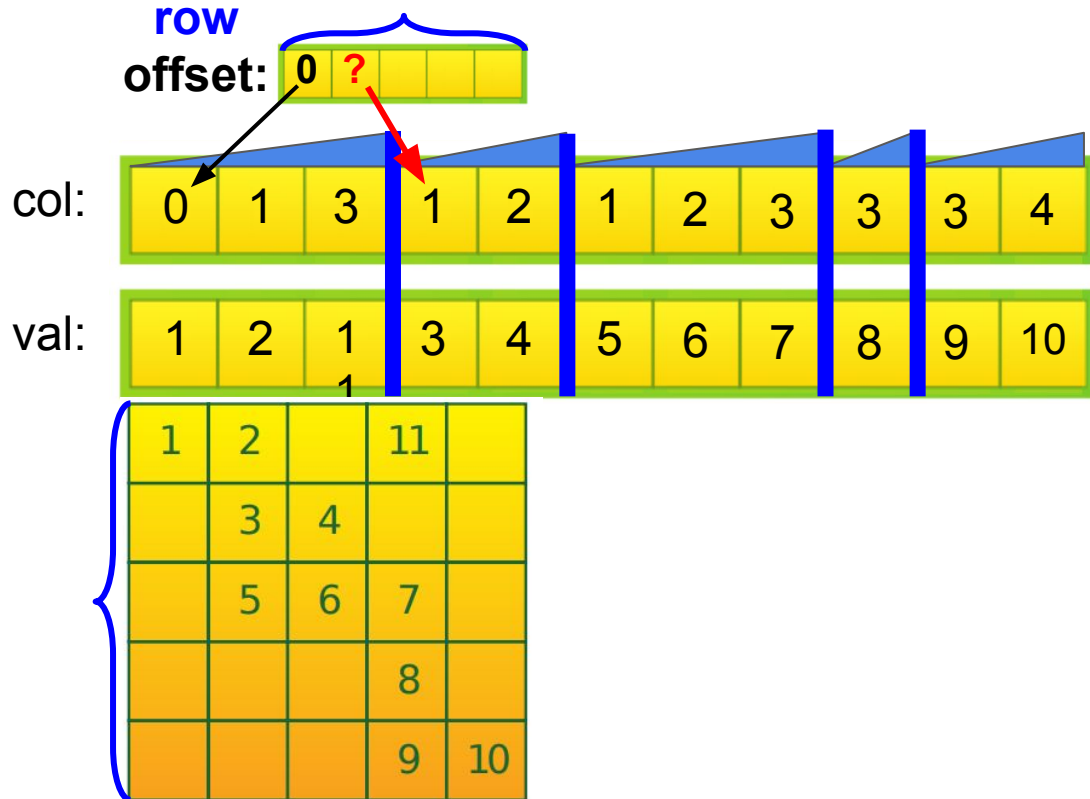


|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

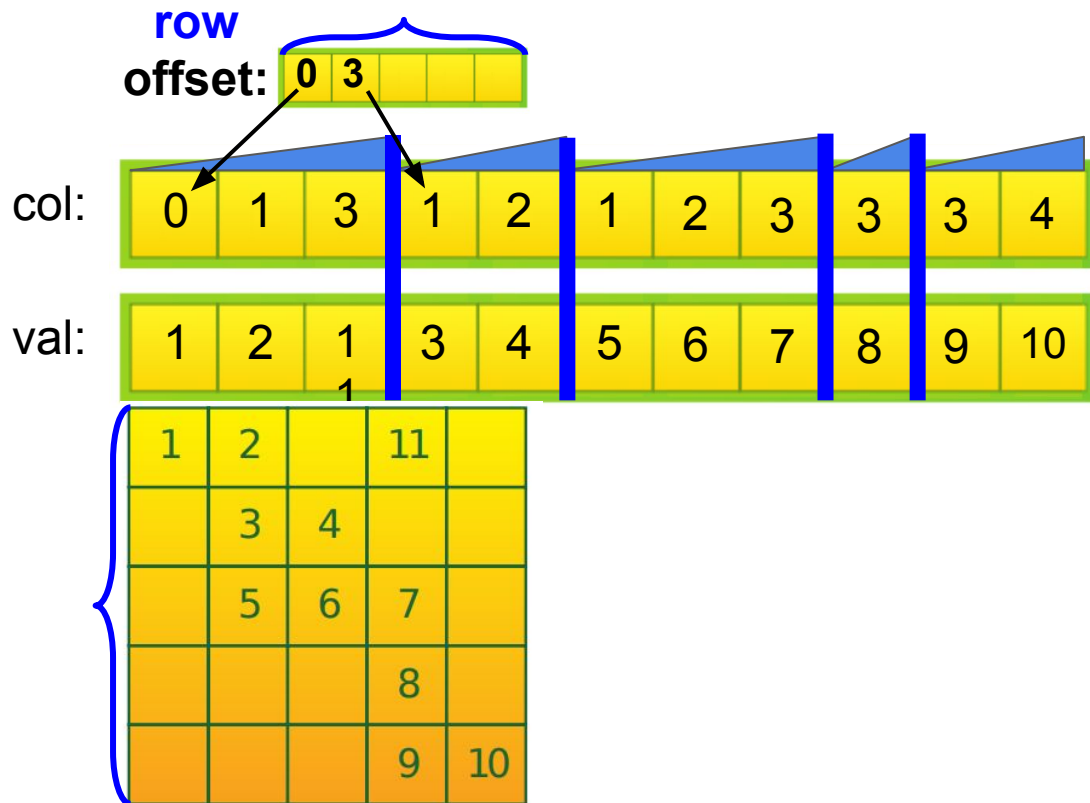
## 2) CSR - Compressed Sparse Row



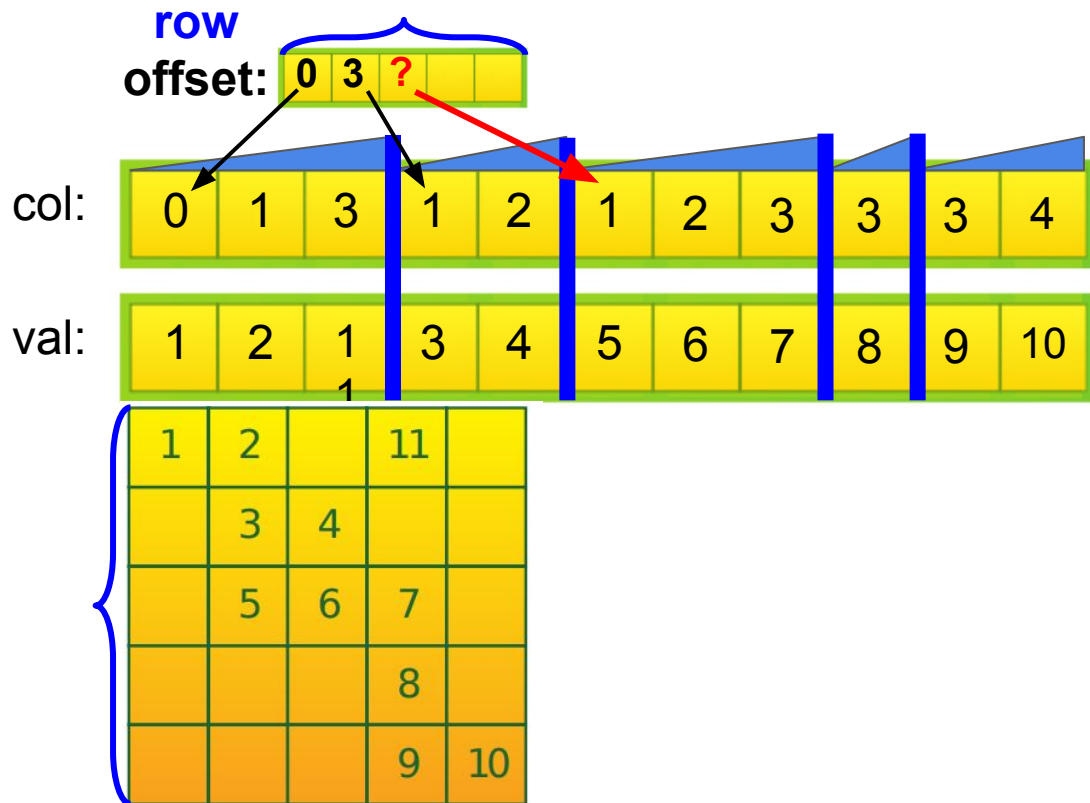
## 2) CSR - Compressed Sparse Row



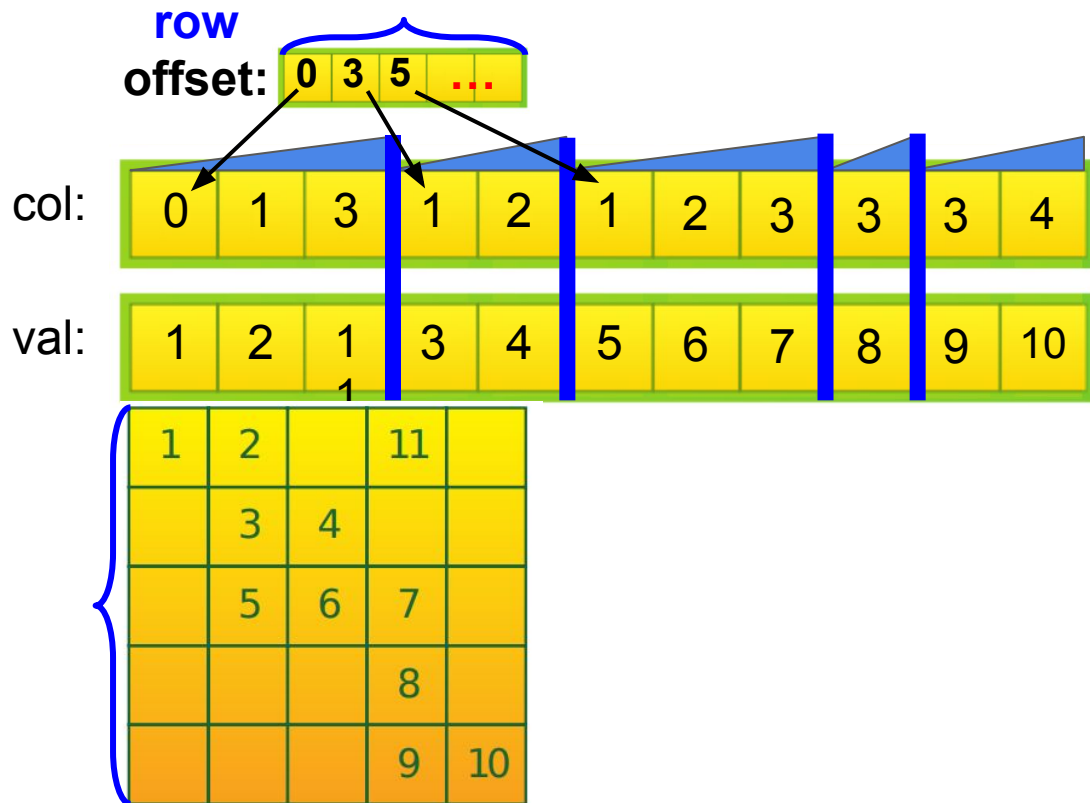
## 2) CSR - Compressed Sparse Row



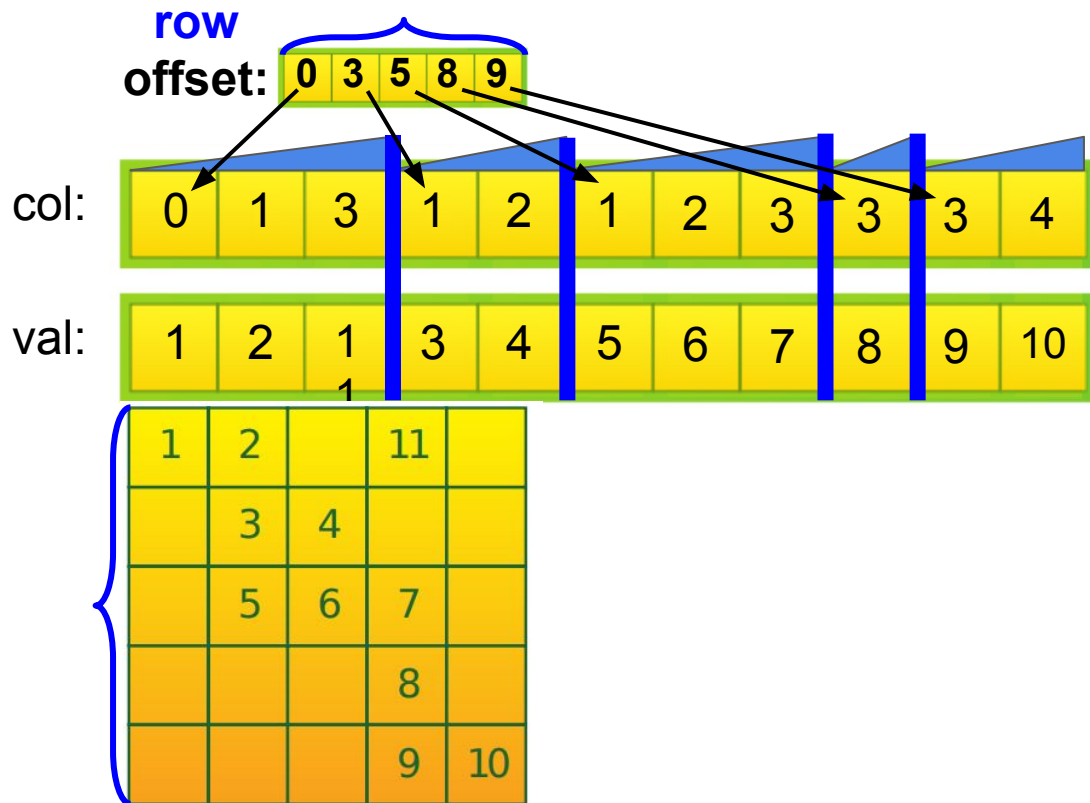
## 2) CSR - Compressed Sparse Row



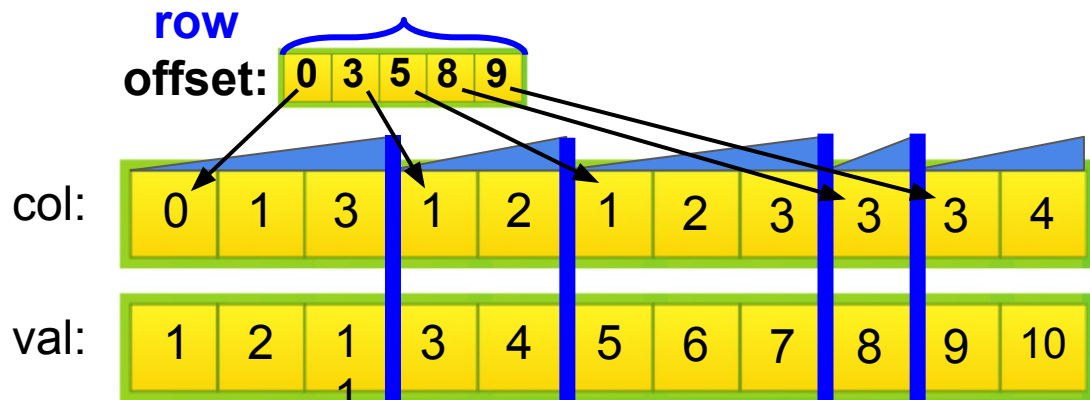
## 2) CSR - Compressed Sparse Row



## 2) CSR - Compressed Sparse Row



## 2) CSR - Compressed Sparse Row



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

Насколько все хорошо?

## 2) CSR - Compressed Sparse Row



Насколько все хорошо?  
- компактность?

## 2) CSR - Compressed Sparse Row

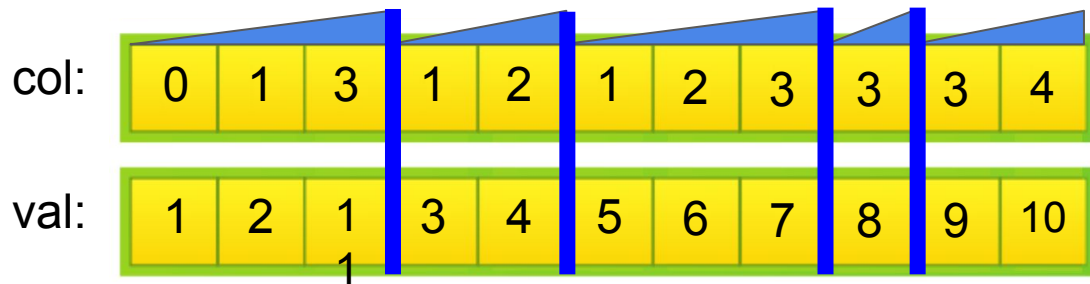


Насколько все хорошо?

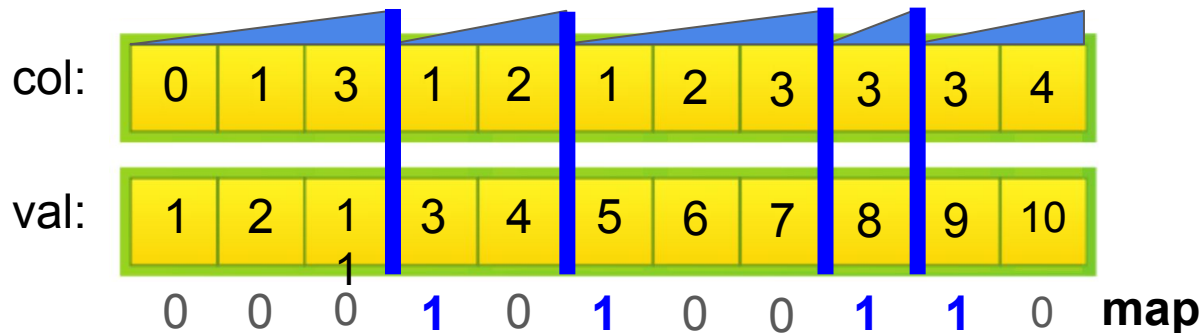
- компактность ( $N + \text{NNZ} + \text{NNZ}$ )

- быстрое создание?

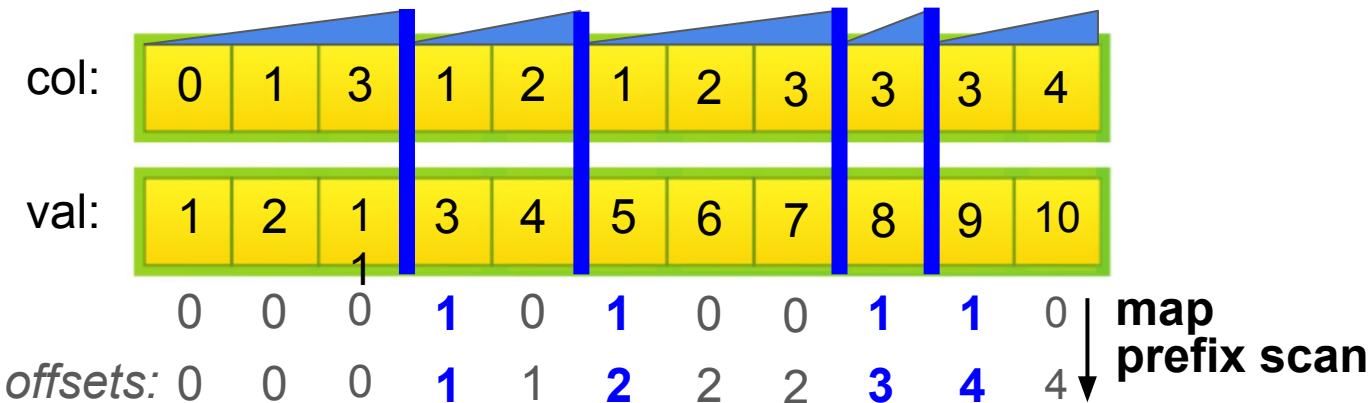
## 2) CSR - Compressed Sparse Row: **быстрое создание?**



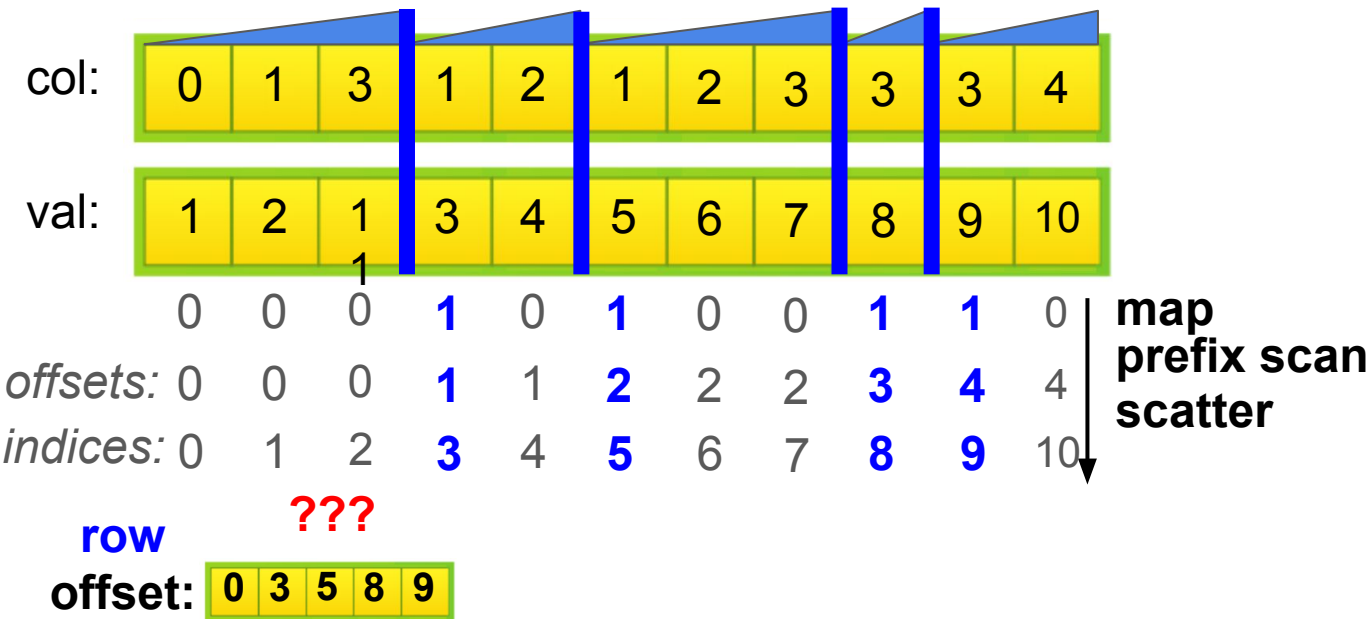
## 2) CSR - Compressed Sparse Row: **быстрое создание?**



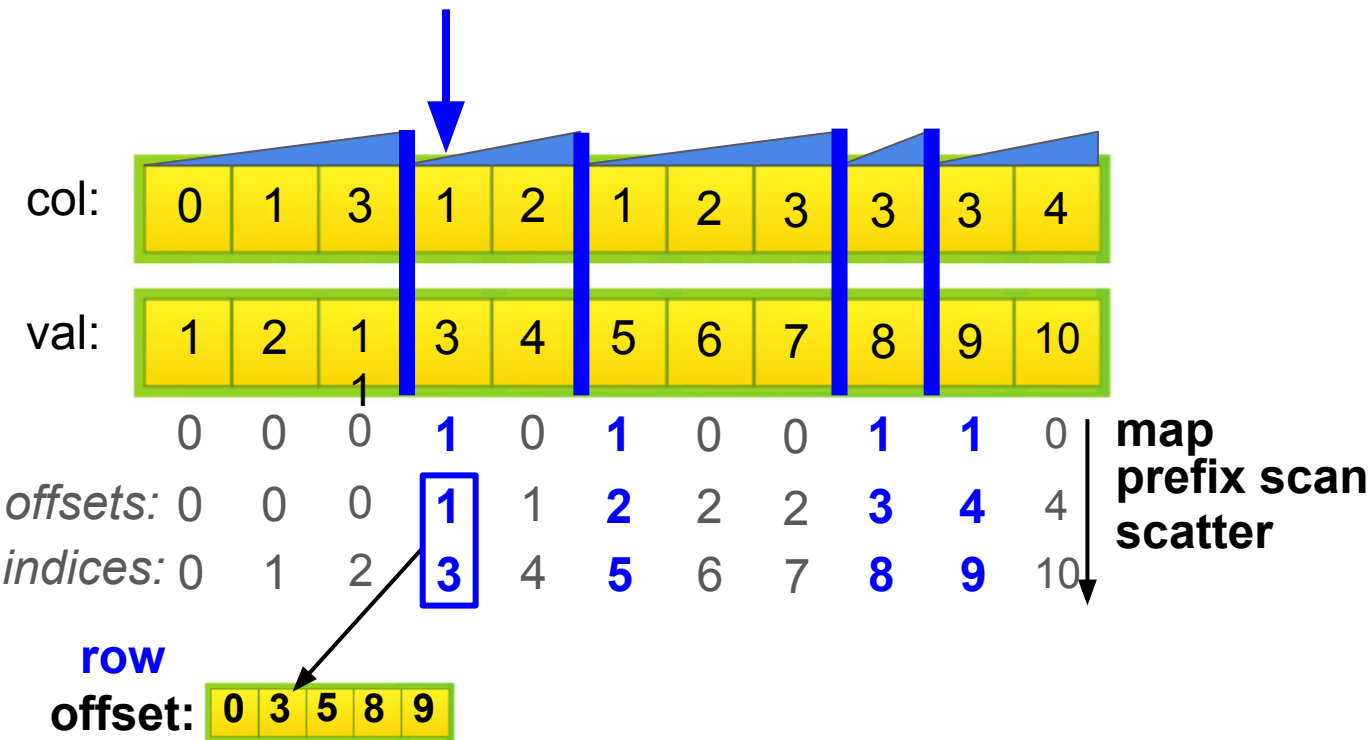
## 2) CSR - Compressed Sparse Row: **быстрое создание?**



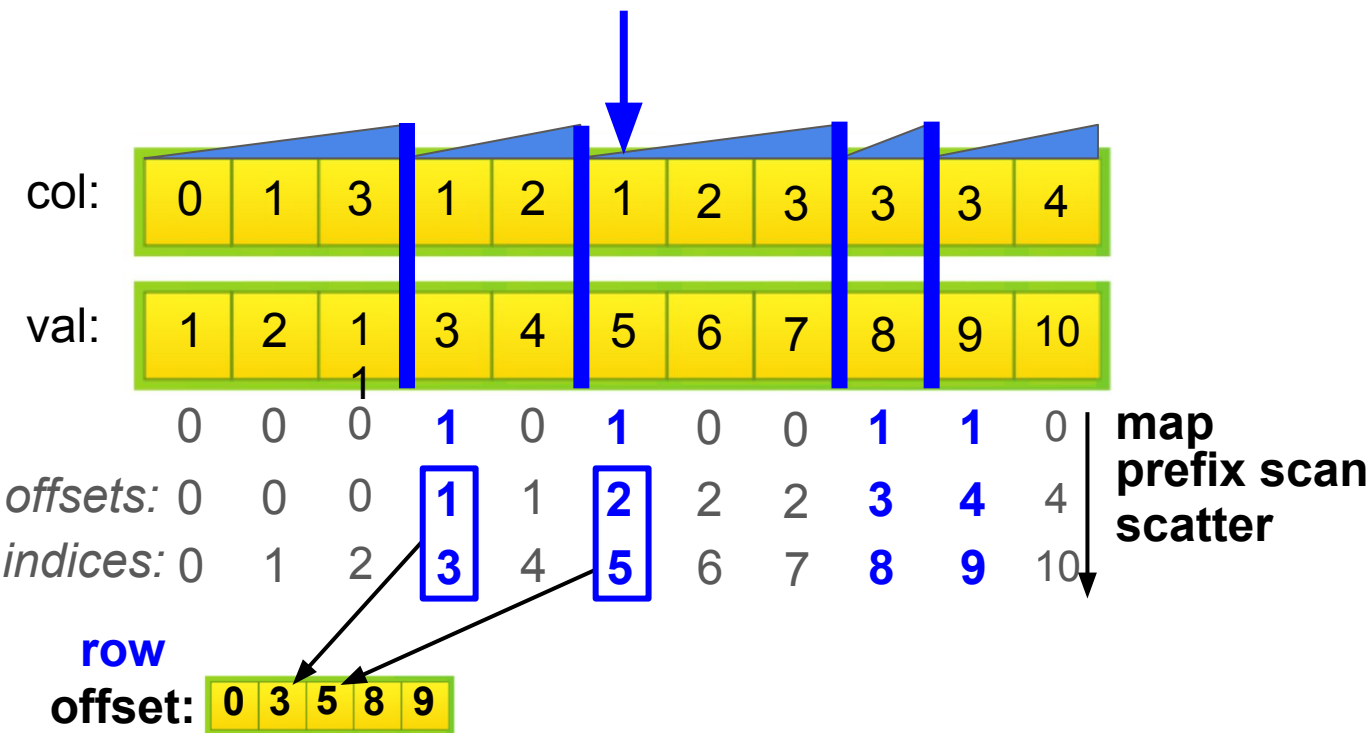
## 2) CSR - Compressed Sparse Row: **быстрое создание?**



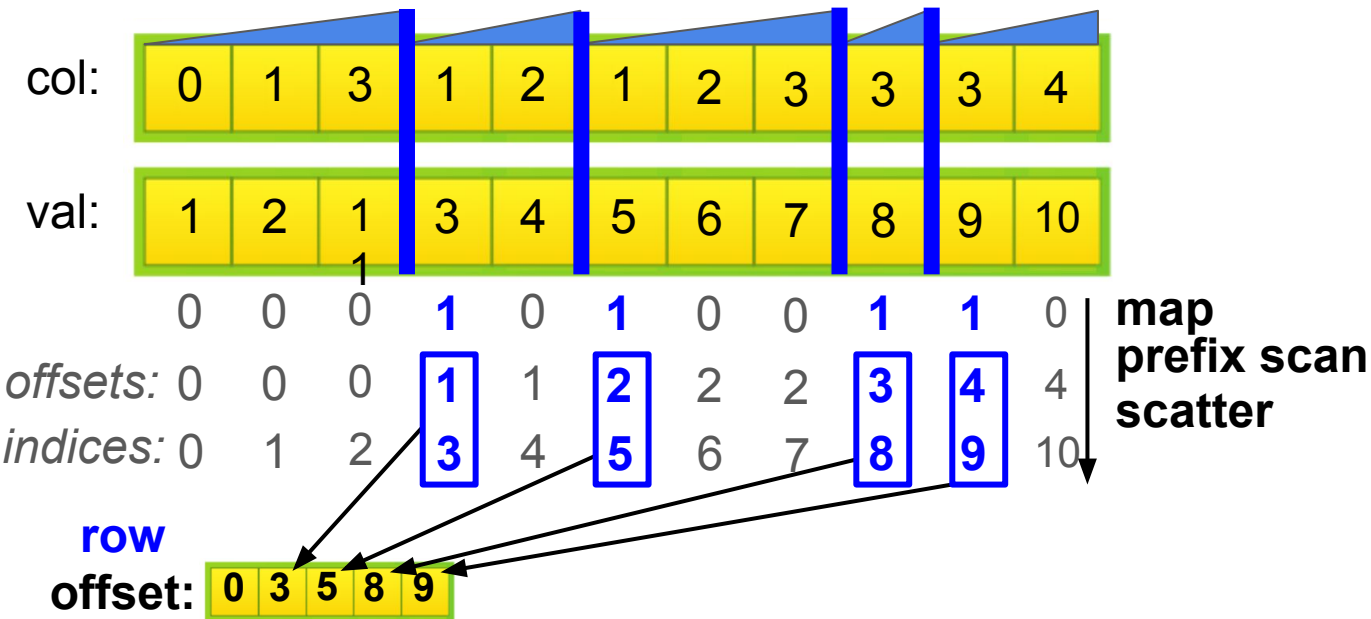
## 2) CSR - Compressed Sparse Row: **быстрое создание?**



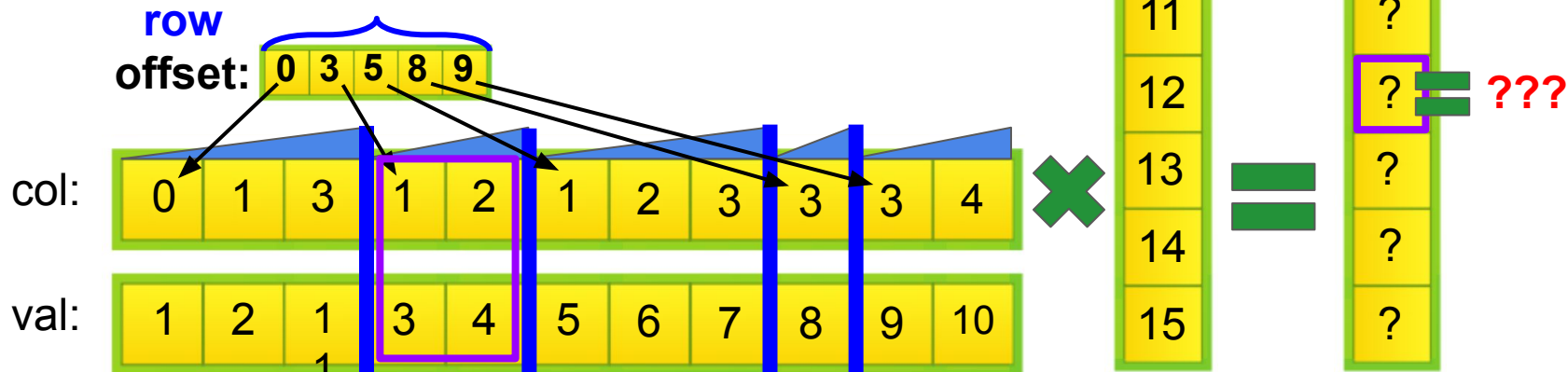
## 2) CSR - Compressed Sparse Row: **быстрое создание?**



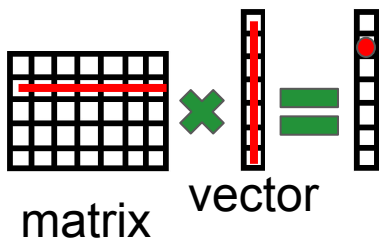
## 2) CSR - Compressed Sparse Row: **быстрое создание?**



## 2) CSR - Compressed Sparse Row



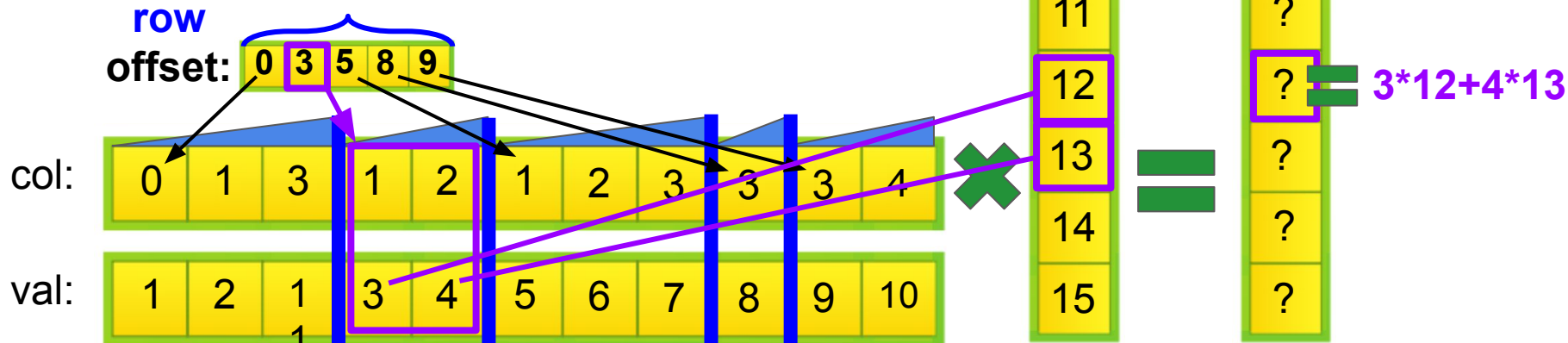
|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |



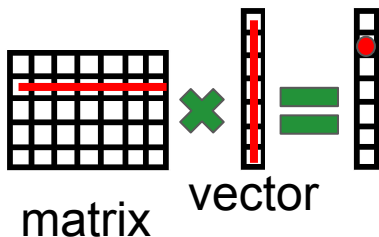
**Насколько все хорошо?**

- компактность ( $N + NNZ + NNZ$ )
- быстрое создание
- **быстрое умножение?**
- что делает WorkGroup?**

## 2) CSR - Compressed Sparse Row



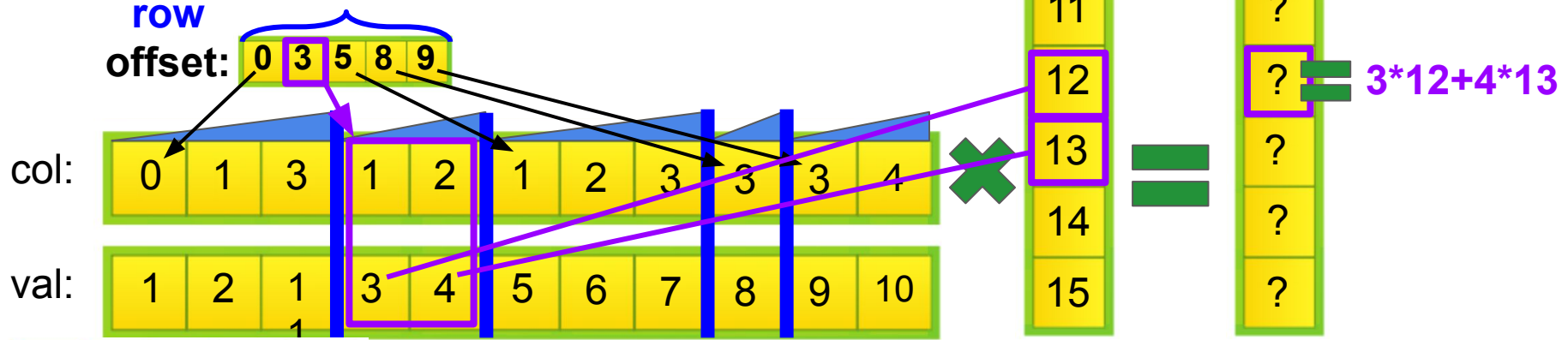
|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |



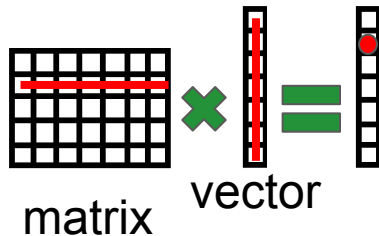
**Насколько все хорошо?**

- компактность ( $N + NNZ + NNZ$ )
- быстрое создание
- **быстрое умножение?**
- что делает WorkGroup?**

## 2) CSR - Compressed Sparse Row



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

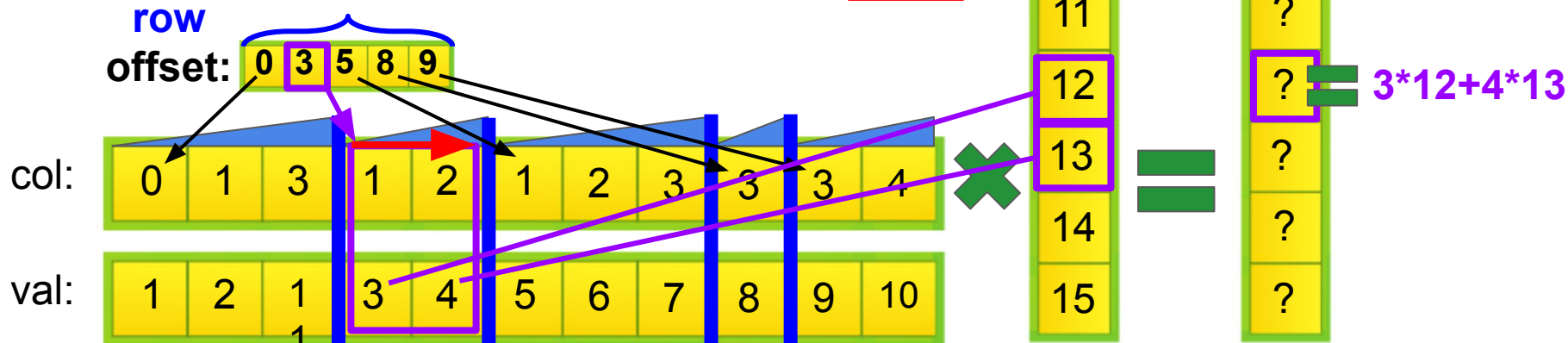


**Насколько все хорошо?**

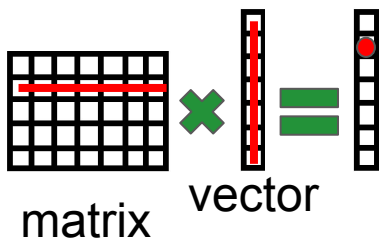
- компактность ( $N + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение (лучше чем **COO**)

Почему именно по рядам сжимаем?

## 2) CSR - Compressed Sparse Row



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |



Насколько все хорошо?

- компактность ( $N + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение (лучше чем COO)

### 3) **CSC** - Compressed **S**parsе **C**olumn

Такие же **Non-Zero Values** как в **CSR**?

val:

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | 2 | 1 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|

1

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) **CSC** - Compressed Sparse Column

Такие же **Non-Zero Values** как в **CSR**?

val:



|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) - Compressed Sparse Column

Как выглядит?

val:

|   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|
| ? | ? | ? | ? | ? | ? | ? | ? | ? | ? | ? |
|---|---|---|---|---|---|---|---|---|---|---|

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) CSC - Compressed Sparse Column

val:

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | 2 | 3 | 5 | 4 | 6 | 1 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|

1

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

### 3) **CSC** - Compressed Sparse **C**olumn

Как выглядит **row**?

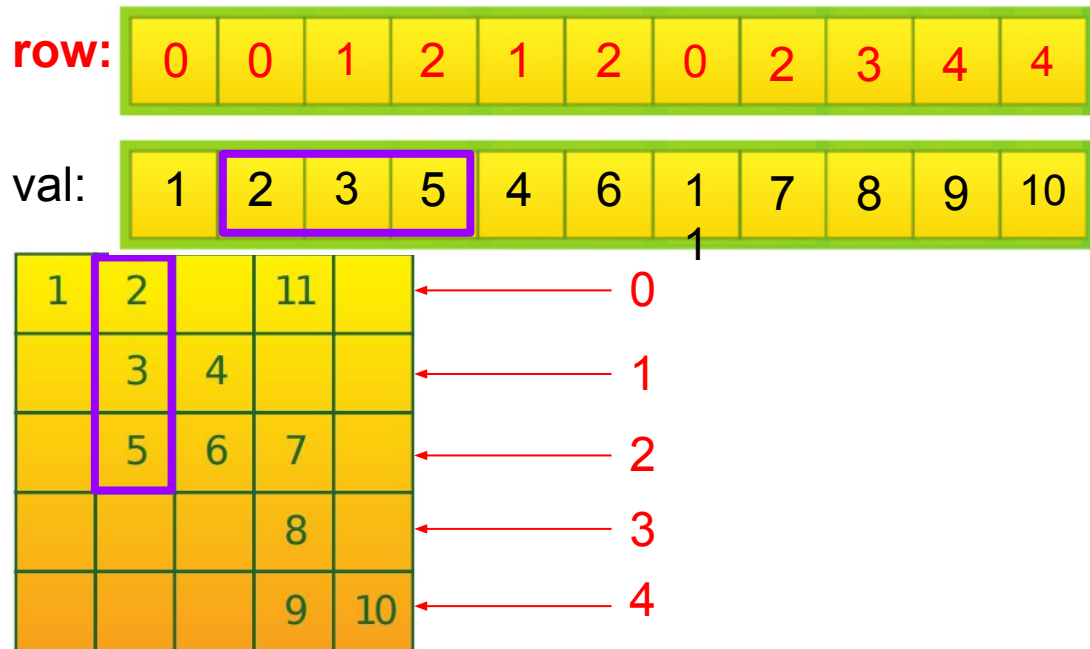
val: 

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | 2 | 3 | 5 | 4 | 6 | 1 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|

|   |   |   |    |    |     |
|---|---|---|----|----|-----|
| 1 | 2 |   | 11 |    | ← 0 |
|   | 3 | 4 |    |    | ← 1 |
|   | 5 | 6 | 7  |    | ← 2 |
|   |   |   | 8  |    | ← 3 |
|   |   |   | 9  | 10 | ← 4 |

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) CSC - Compressed Sparse Column



### 3) CSC - Compressed Sparse Column

Как выглядит **column offset**?

row:

|   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 | 2 | 0 | 2 | 3 | 4 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|

val:

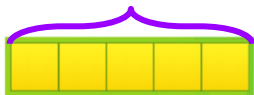
|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | 2 | 3 | 5 | 4 | 6 | 1 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

### 3) CSC - Compressed Sparse Column

column

offset:



Как выглядит column offset?

row:

|   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 | 2 | 0 | 2 | 3 | 4 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|

val:

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | 2 | 3 | 5 | 4 | 6 | 1 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|

1

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) - Compressed Sparse Column

column

offset:

0 1 4 6 10

row:

|   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 | 2 | 0 | 2 | 3 | 4 | 4 |
|---|---|---|---|---|---|---|---|---|---|---|

val:

|   |   |   |   |   |   |   |   |   |   |    |
|---|---|---|---|---|---|---|---|---|---|----|
| 1 | 2 | 3 | 5 | 4 | 6 | 1 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|----|

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) - Compressed Sparse Column

column

offset:

0 1 4 6 10

row:

0 0 1 2 1 2 0 2 3 4 4

val:

1 2 3 5 4 6 1 7 8 9 10

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

Насколько все хорошо?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) - Compressed Sparse Column

column

offset:

0 1 4 6 10

row:

0 0 1 2 1 2 0 2 3 4 4

val:

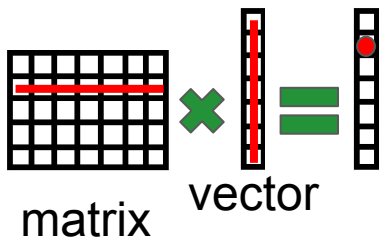
1 2 3 5 4 6 1 7 8 9 10



|    |   |
|----|---|
| 11 | ? |
| 12 | ? |
| 13 | ? |
| 14 | ? |
| 15 | ? |

что из матрицы  
нужно?

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |



Насколько все хорошо?

- компактность ( $N + NNZ + NNZ$ )
- быстрое создание
- быстрое умножение?

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) - Compressed Sparse Column

column

offset:

0 1 4 6 10

row:

0 0 1 2 1 2 0 2 3 4 4

val:

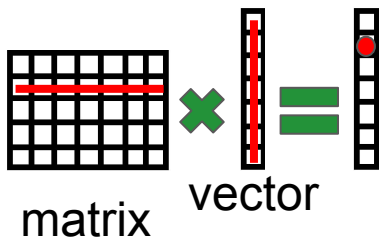
1 2 3 5 4 6 1 7 8 9 10



|    |   |
|----|---|
| 11 | ? |
| 12 | ? |
| 13 | ? |
| 14 | ? |
| 15 | ? |

что из матрицы  
нужно?

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |



Насколько все хорошо?

- компактность ( $N + NNZ + NNZ$ )
- быстрое создание
- **быстрое умножение?**

### 3) - Compressed Sparse Column

column

offset:

0 1 4 6 10

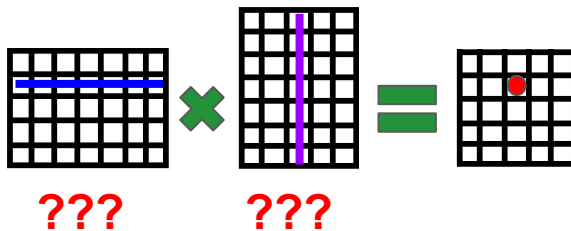
row:

0 0 1 2 1 2 0 2 3 4 4

val:

1 2 3 5 4 6 1 7 8 9 10

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |



Насколько все хорошо?

- компактность ( $N + NNZ + NNZ$ )
- быстрое создание
- **быстрое умножение матриц!**

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

### 3) - Compressed Sparse **Column**

column

offset:

0 1 4 6 10

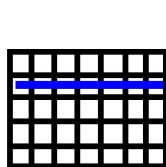
row:

0 0 1 2 1 2 0 2 3 4 4

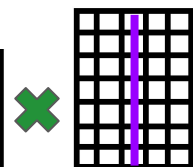
val:

1 2 3 5 4 6 1 7 8 9 10

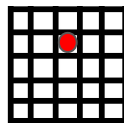
|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |



CSR



CSC



**Насколько все хорошо?**

- компактность ( $N + NNZ + NNZ$ )
- быстрое создание
- **быстрое умножение матриц!**

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

П е р е р ы в !

## Глава 2:

ELLpack, DIagonal, HYBrid

## 4) ELL - ELLPACK

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

Сколько максимум NNZ элементов в ряде?

## 4) ELL - ELLPACK

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

values

|   |   |    |
|---|---|----|
| 1 | 2 | 11 |
|   |   |    |
|   |   |    |
|   |   |    |
|   |   |    |

## 4) ELL - ELLPACK

| column |   |   |    |    | values |   |   |
|--------|---|---|----|----|--------|---|---|
| 1      | 2 |   | 11 |    | ?      | ? | ? |
|        | 3 | 4 |    |    |        |   |   |
|        | 5 | 6 | 7  |    |        |   |   |
|        |   |   | 8  |    |        |   |   |
|        |   |   | 9  | 10 |        |   |   |

## 4) ELL - ELLPACK

|   |   |   |    |    | column |   |   | values |   |    |
|---|---|---|----|----|--------|---|---|--------|---|----|
| 1 | 2 |   | 11 |    | 0      | 1 | 3 | 1      | 2 | 11 |
|   | 3 | 4 |    |    |        |   |   |        |   |    |
|   | 5 | 6 | 7  |    |        |   |   |        |   |    |
|   |   |   | 8  |    |        |   |   |        |   |    |
|   |   |   | 9  | 10 |        |   |   |        |   |    |

## 4) ELL - ELLPACK

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

column

|   |   |   |
|---|---|---|
| 0 | 1 | 3 |
| 1 | 2 |   |
| 1 | 2 | 3 |
| 3 |   |   |
| 3 | 4 |   |

values

|   |    |    |
|---|----|----|
| 1 | 2  | 11 |
| 3 | 4  |    |
| 5 | 6  | 7  |
| 8 |    |    |
| 9 | 10 |    |

**Насколько все хорошо?**

## 4) ELL - ELLPACK

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

column

|   |   |   |
|---|---|---|
| 0 | 1 | 3 |
| 1 | 2 |   |
| 1 | 2 | 3 |
| 3 |   |   |
| 3 | 4 |   |

values

|   |    |    |
|---|----|----|
| 1 | 2  | 11 |
| 3 | 4  |    |
| 5 | 6  | 7  |
| 8 |    |    |
| 9 | 10 |    |

Насколько все хорошо?

- компактность ( $N \cdot K + N \cdot K$ ), хоть и хуже чем  $N + NNZ + NNZ$
- зато ???
- быстрое создание
- быстрое умножение

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

## 4) ELL - ELLPACK

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

column

|   |   |   |
|---|---|---|
| 0 | 1 | 3 |
| 1 | 2 |   |
| 1 | 2 | 3 |
| 3 |   |   |
| 3 | 4 |   |

values

|   |    |    |
|---|----|----|
| 1 | 2  | 11 |
| 3 | 4  |    |
| 5 | 6  | 7  |
| 8 |    |    |
| 9 | 10 |    |

Насколько все хорошо?

- компактность ( $N \cdot K + N \cdot K$ ), хоть и хуже чем  $N + NNZ + NNZ$
- зато **равномерность рабочей нагрузки** если везде  $\sim K$
- быстрое создание
- быстрое умножение

Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

## 5) DIA - DIagonal идеи?

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

## 5) DIA - DIagonal

|          |   |   |    |    |
|----------|---|---|----|----|
| values   |   |   |    |    |
| 1        | 2 |   | 11 |    |
|          | 3 | 4 |    |    |
|          | 5 | 6 | 7  |    |
|          |   |   | 8  |    |
|          |   |   | 9  | 10 |
| diagonal |   |   |    |    |
| -1       |   |   |    |    |

## 5) DIA - DIagonal

values

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

|   |    |  |  |
|---|----|--|--|
|   | 1  |  |  |
|   | 3  |  |  |
| 5 | 6  |  |  |
|   | 7  |  |  |
| 9 | 10 |  |  |

diagonal

|    |   |  |  |
|----|---|--|--|
| -1 | 0 |  |  |
|----|---|--|--|

## 5) DIA - DIagonal

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

values

|   |    |   |  |
|---|----|---|--|
|   | 1  | 2 |  |
|   | 3  | 4 |  |
| 5 | 6  | 7 |  |
|   | 7  |   |  |
| 9 | 10 |   |  |

diagonal

|    |   |    |  |
|----|---|----|--|
| -1 | 0 | +1 |  |
|----|---|----|--|

## 5) DIA - DIAGONAL

|          |   |    |    |    |
|----------|---|----|----|----|
| values   |   |    |    |    |
| 1        | 2 |    | 11 |    |
|          | 3 | 4  |    |    |
|          | 5 | 6  | 7  |    |
|          |   |    | 8  |    |
|          |   |    | 9  | 10 |
| diagonal |   |    |    |    |
| -1       | 0 | +1 | +3 |    |

## 5) DIA - DIAGONAL

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

values

|   |    |   |    |  |
|---|----|---|----|--|
|   | 1  | 2 | 11 |  |
|   | 3  | 4 |    |  |
| 5 | 6  | 7 |    |  |
|   | 7  |   |    |  |
| 9 | 10 |   |    |  |

Насколько все хорошо?

diagonal

|    |   |    |    |
|----|---|----|----|
| -1 | 0 | +1 | +3 |
|----|---|----|----|

## 5) DIA - DIAGONAL

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

values

|   |    |   |    |  |
|---|----|---|----|--|
|   | 1  | 2 | 11 |  |
|   | 3  | 4 |    |  |
| 5 | 6  | 7 |    |  |
|   | 7  |   |    |  |
| 9 | 10 |   |    |  |

diagonal

|    |   |    |    |
|----|---|----|----|
| -1 | 0 | +1 | +3 |
|----|---|----|----|

**Насколько все хорошо?**

- компактность ( $D \cdot N + D$ ), хуже чем  $2 \cdot \text{NNZ}$
- зато **равномерность** рабочей нагрузки
- быстрое умножение

## 5) DIA - DIAGONAL

|   |   |   |    |    |
|---|---|---|----|----|
| 1 | 2 |   | 11 |    |
|   | 3 | 4 |    |    |
|   | 5 | 6 | 7  |    |
|   |   |   | 8  |    |
|   |   |   | 9  | 10 |

values

|   |    |   |    |  |
|---|----|---|----|--|
|   | 1  | 2 | 11 |  |
|   | 3  | 4 |    |  |
| 5 | 6  | 7 |    |  |
|   | 7  |   |    |  |
| 9 | 10 |   |    |  |

diagonal

|    |   |    |    |
|----|---|----|----|
| -1 | 0 | +1 | +3 |
|----|---|----|----|

Насколько все хорошо?

- компактность ( $\mathbf{D} \times \mathbf{N} + \mathbf{D}$ ), хуже чем  $2 \times \mathbf{NNZ}$
- зато **равномерность рабочей нагрузки**
- быстрое умножение

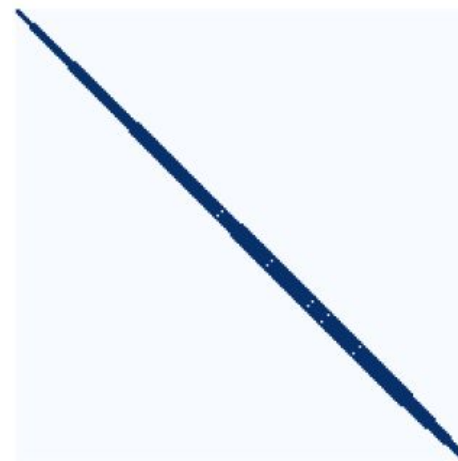
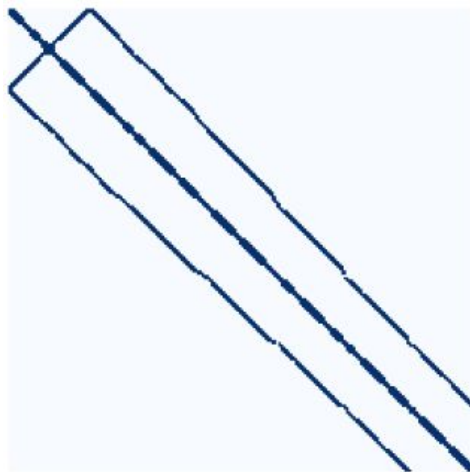
| Format | Structure               | Values                           |
|--------|-------------------------|----------------------------------|
| Dense  | —                       | $N \times N$                     |
| COO    | $2 \times \mathbf{NNZ}$ | $\mathbf{NNZ}$                   |
| CSR    | $N + 1 + \mathbf{NNZ}$  | $\mathbf{NNZ}$                   |
| ELL    | $M \times N$            | $M \times N$                     |
| DIA    | $\mathbf{D}$            | $\mathbf{D} \times \mathbf{N}_D$ |

## 5) DIA - DIAgonal

Когда хороши?

| Format | Structure      | Values         |
|--------|----------------|----------------|
| Dense  | –              | $N \times N$   |
| COO    | $2 \times NNZ$ | $NNZ$          |
| CSR    | $N + 1 + NNZ$  | $NNZ$          |
| ELL    | $M \times N$   | $M \times N$   |
| DIA    | $D$            | $D \times N_D$ |

## 5) DIA - DIAgonal

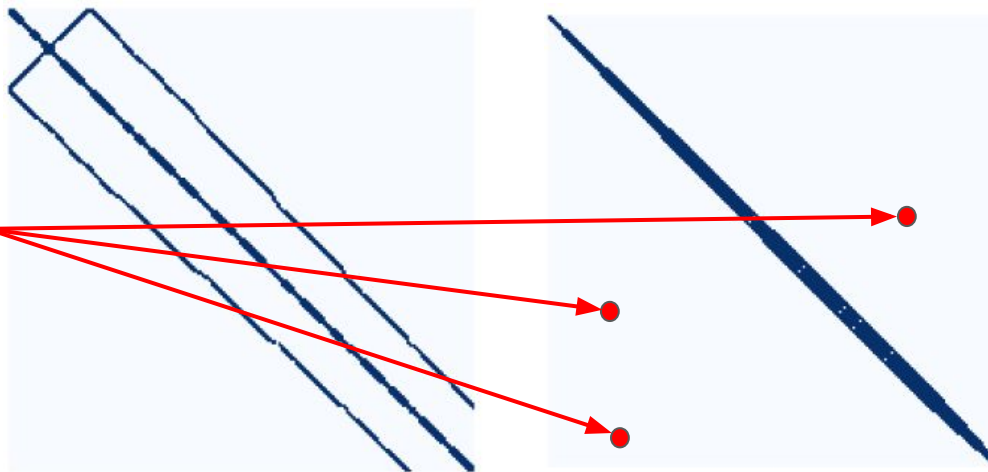


| Format | Structure      | Values         |
|--------|----------------|----------------|
| Dense  | —              | $N \times N$   |
| COO    | $2 \times NNZ$ | $NNZ$          |
| CSR    | $N + 1 + NNZ$  | $NNZ$          |
| ELL    | $M \times N$   | $M \times N$   |
| DIA    | $D$            | $D \times N_D$ |

Когда хороши?

## 5) DIA - DIAgonal

А если небольшая доля выбросов?

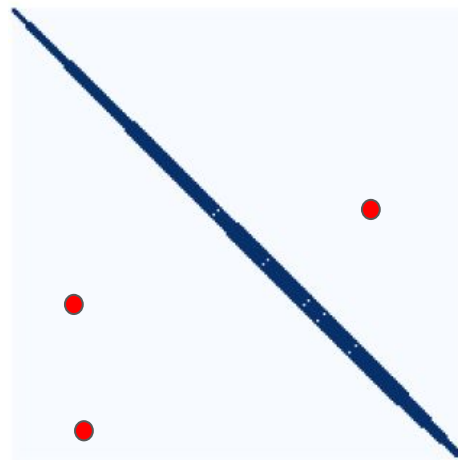


| Format | Structure      | Values         |
|--------|----------------|----------------|
| Dense  | —              | $N \times N$   |
| COO    | $2 \times NNZ$ | $NNZ$          |
| CSR    | $N + 1 + NNZ$  | $NNZ$          |
| ELL    | $M \times N$   | $M \times N$   |
| DIA    | $D$            | $D \times N_D$ |

Когда хороши?

6) **HYB = ELL/DIA + COO**

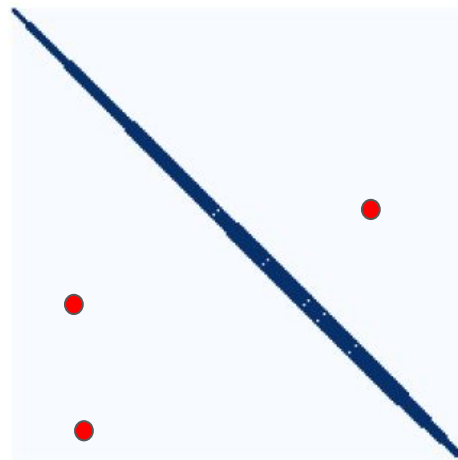
**A = B** + **C** (для выбросов)



6) HYB = ELL/DIA + COO

$A = B + C$  (для выбросов)

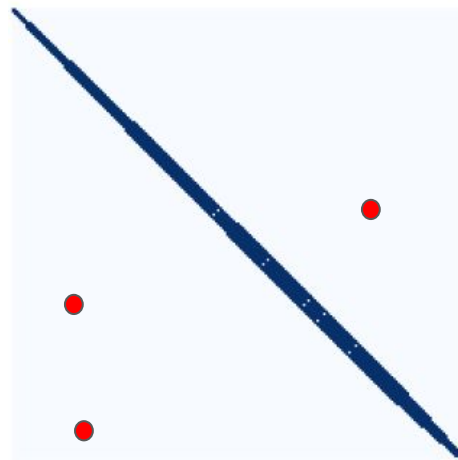
Как делать  $Axv$ ?



## 6) HYB = ELL/DIA + COO

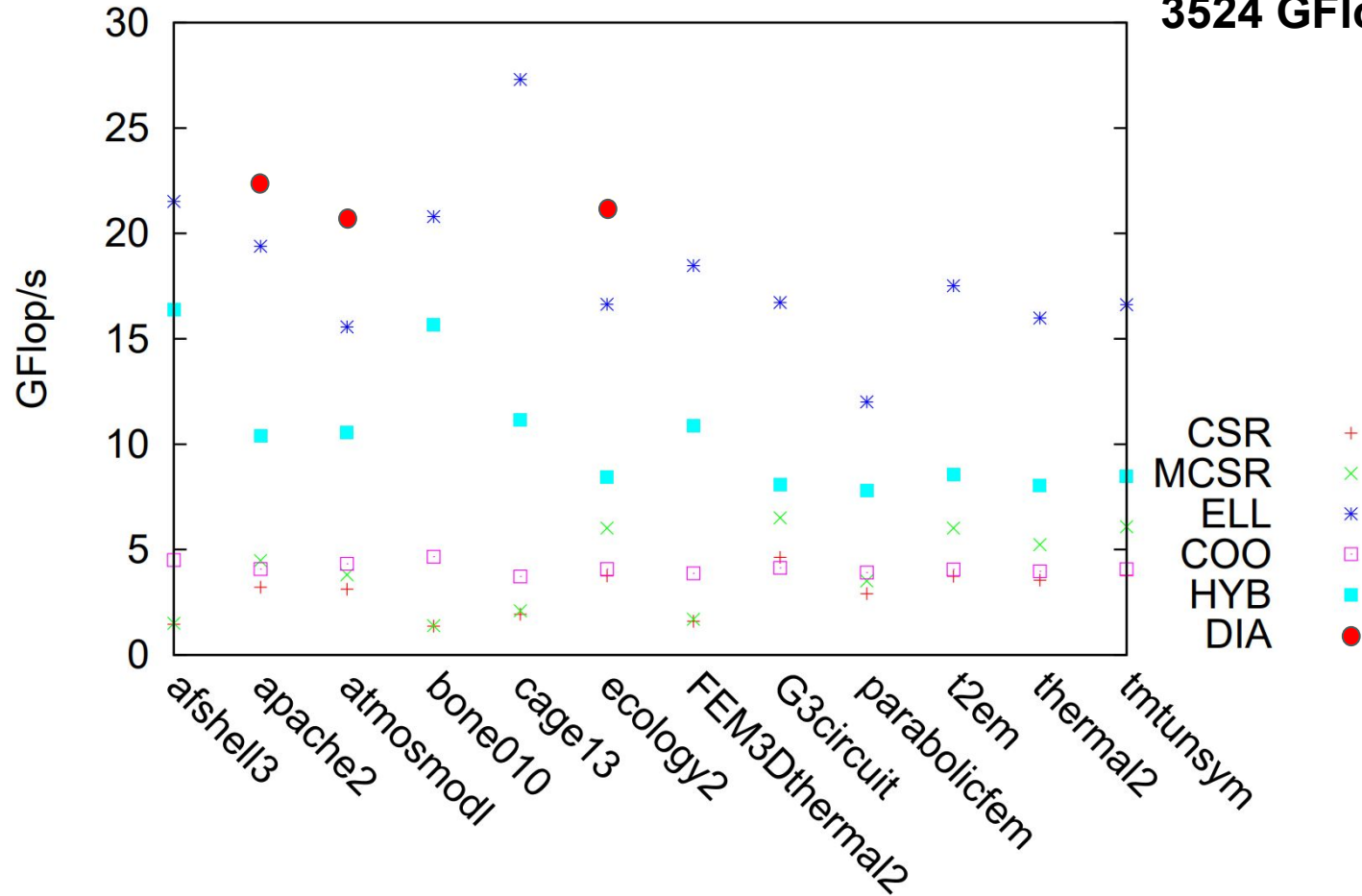
$$A = B + C \text{ (для выбросов)}$$

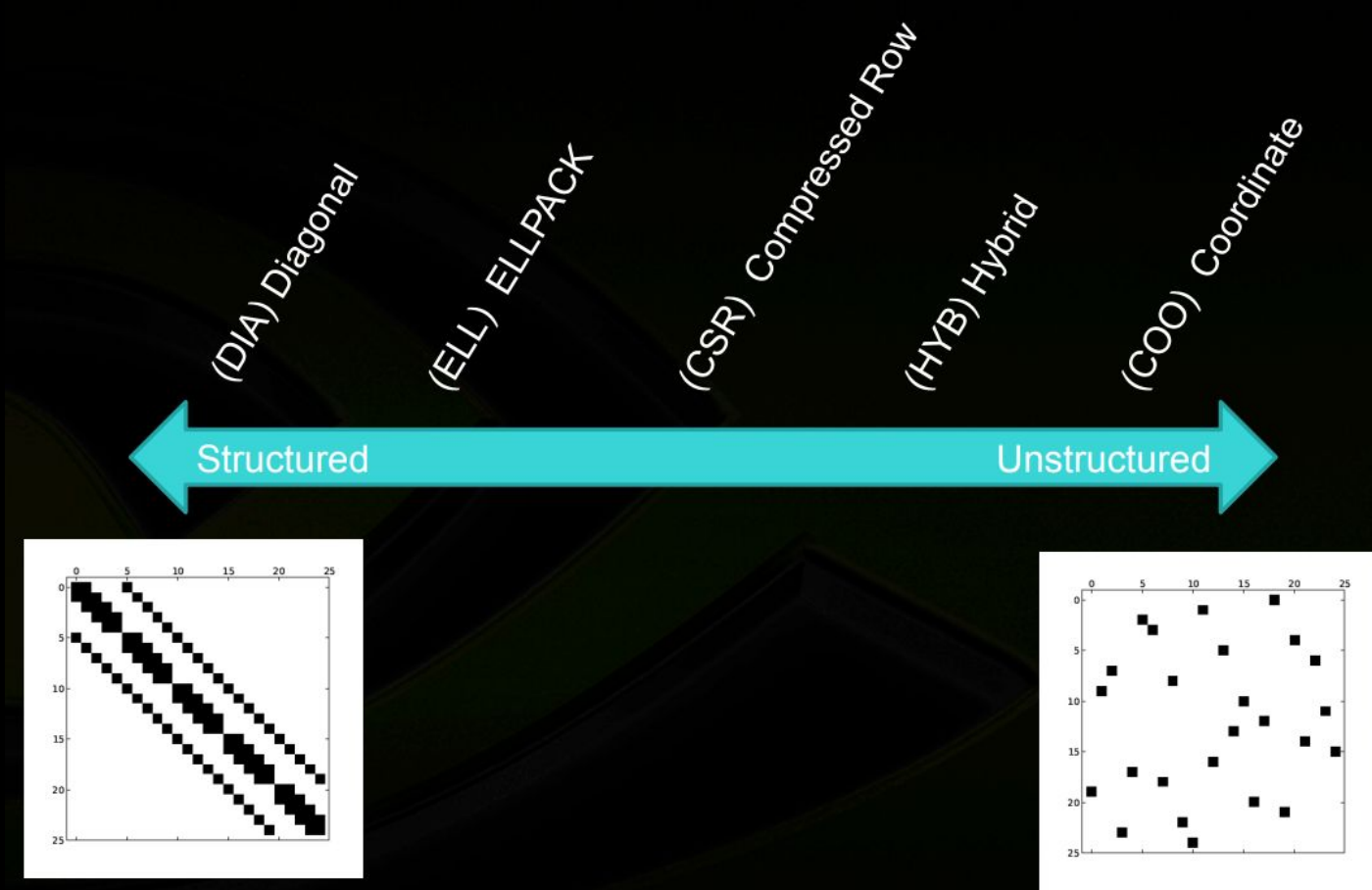
$$Axv = Bxv + Cxv$$



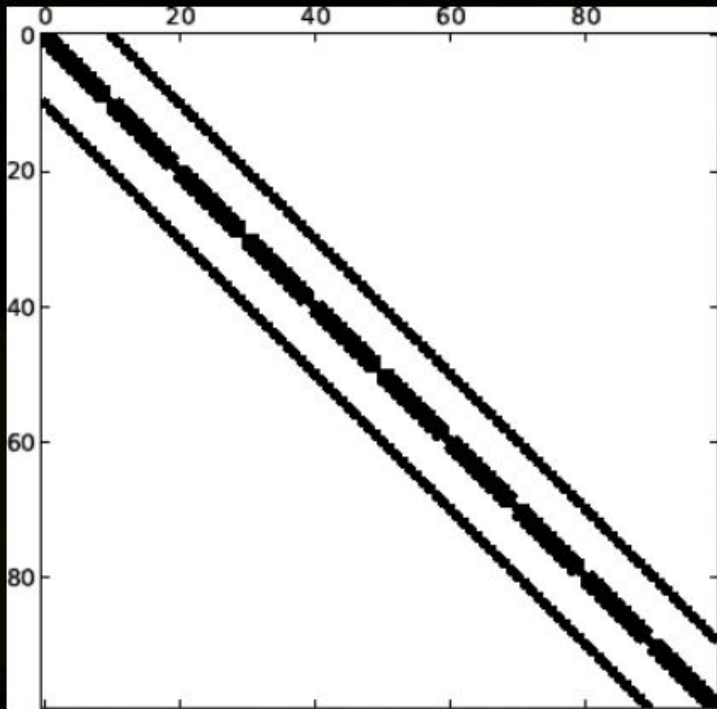
# Sparse Matrix-Vector Multiplication on K20 GPU

**208 Gb/s bandwidth**  
**3524 GFlop/s**





# Structured Mesh

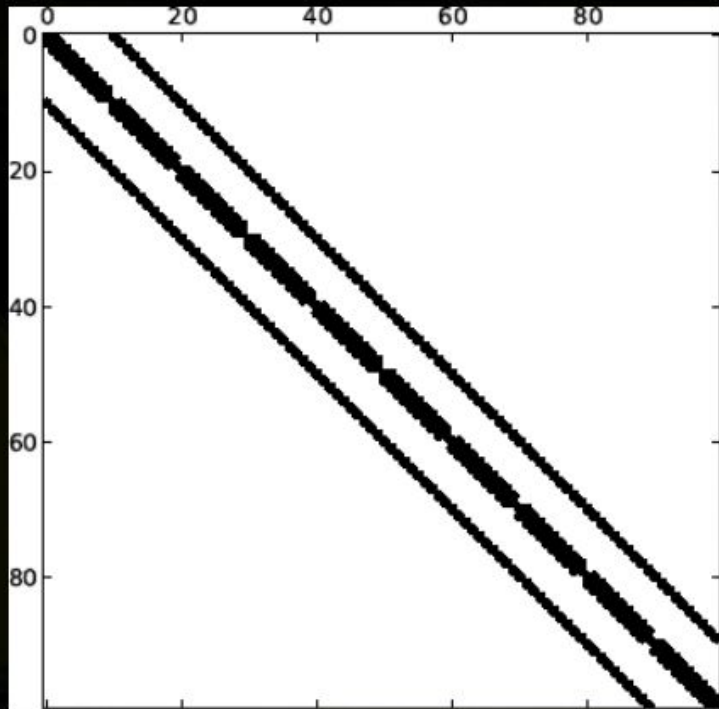


Matrix

| Format | float           | double |
|--------|-----------------|--------|
| COO    | Кто компактнее? |        |
| CSR    |                 |        |
| DIA    |                 |        |
| ELL    |                 |        |
| HYB    |                 |        |

Bytes per Nonzero Entry

# Structured Mesh

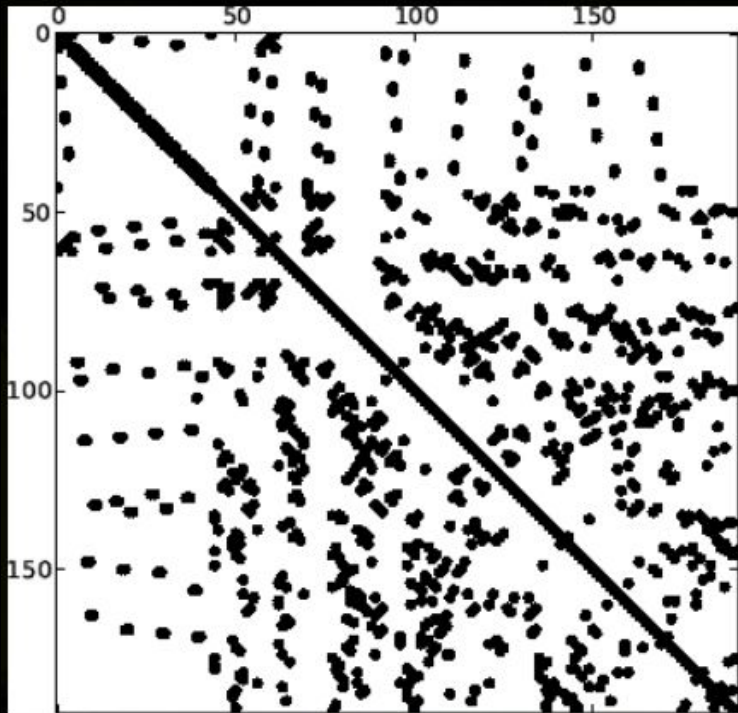


Matrix

| Format | float | double |
|--------|-------|--------|
| COO    | 12.00 | 16.00  |
| CSR    | 8.45  | 12.45  |
| DIA    | 4.05  | 8.10   |
| ELL    | 8.11  | 12.16  |
| HYB    | 8.11  | 12.16  |

Bytes per Nonzero Entry

# Unstructured Mesh

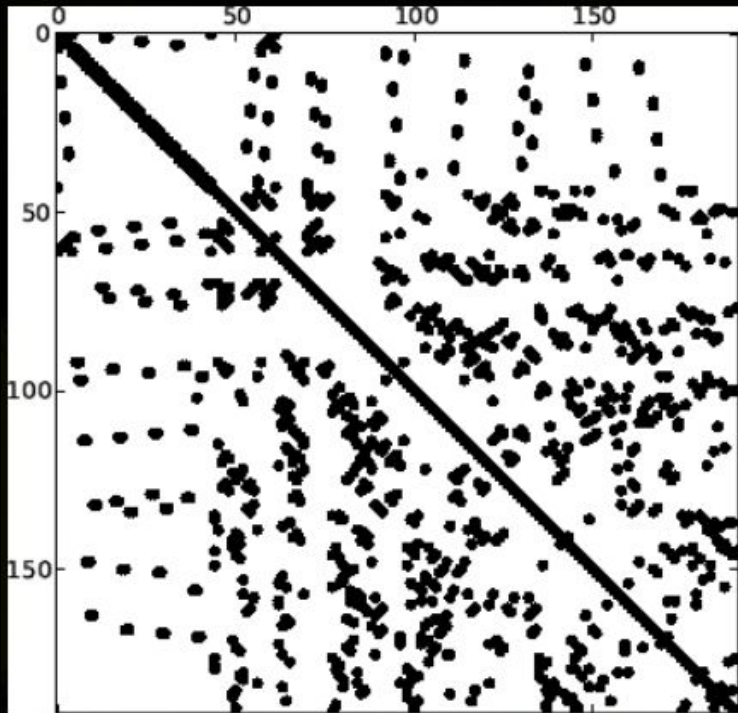


Matrix

| Format | float           | double |
|--------|-----------------|--------|
| COO    | Кто компактнее? |        |
| CSR    |                 |        |
| DIA    |                 |        |
| ELL    |                 |        |
| HYB    |                 |        |

Bytes per Nonzero Entry

# Unstructured Mesh

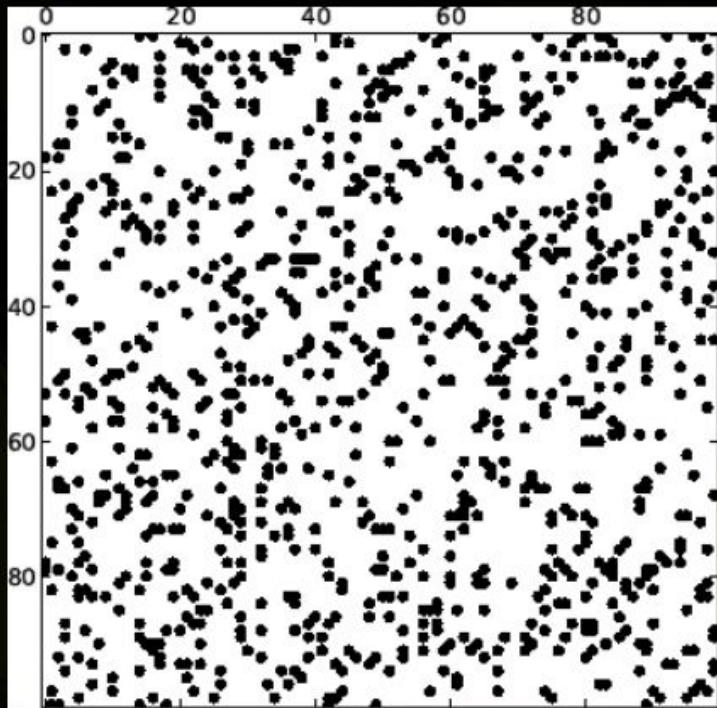


Matrix

| Format | float  | double |
|--------|--------|--------|
| COO    | 12.00  | 16.00  |
| CSR    | 8.62   | 12.62  |
| DIA    | 164.11 | 328.22 |
| ELL    | 11.06  | 16.60  |
| HYB    | 9.00   | 13.44  |

Bytes per Nonzero Entry

# Random

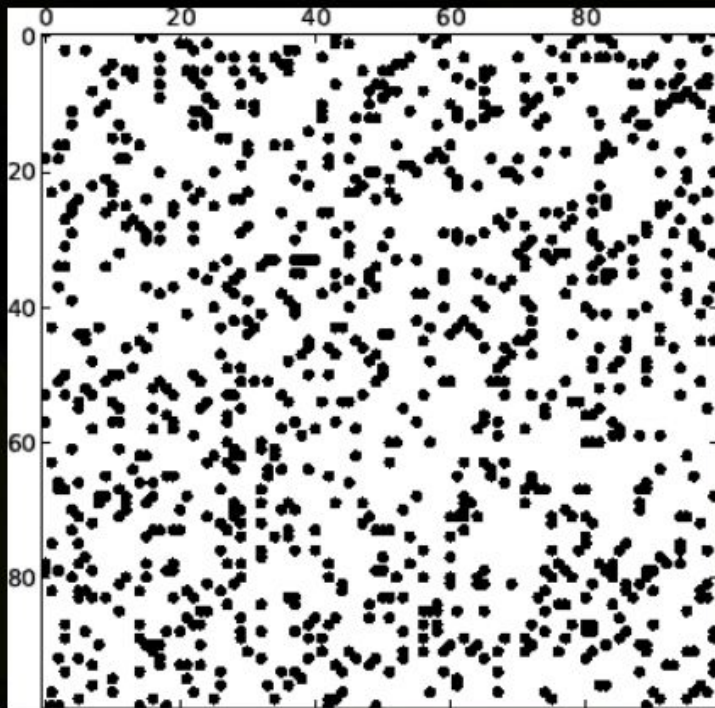


Matrix

| Format | Float           | Double |
|--------|-----------------|--------|
| COO    | Кто компактнее? |        |
| CSR    |                 |        |
| DIA    |                 |        |
| ELL    |                 |        |
| HYB    |                 |        |

Bytes per Nonzero Entry

# Random



Matrix

| Format | Float | Double |
|--------|-------|--------|
| COO    | 12.00 | 16.00  |
| CSR    | 8.42  | 12.42  |
| DIA    | 76.83 | 153.65 |
| ELL    | 14.20 | 21.29  |
| HYB    | 9.60  | 14.20  |

Bytes per Nonzero Entry

Источник: [Sparse Matrix Representations & Iterative Solvers](#)

# Глава 3:

## Системы Непересекающихся Множеств

# Система Непересекающихся Множеств - CHM

Disjoint Set



# Система Непересекающихся Множеств - CHM

Disjoint Set



find(2) = 2

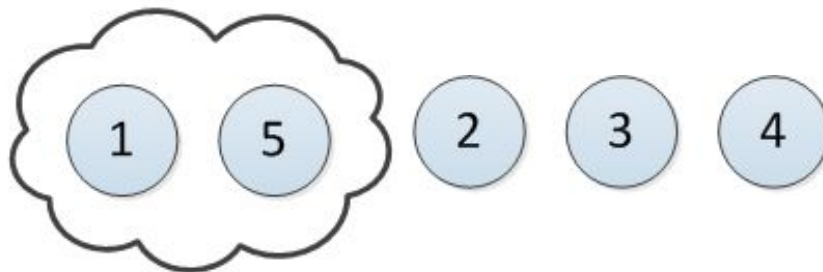
# Система Непересекающихся Множеств - CHM

Disjoint Set



$\text{find}(2) = 2$

$\text{union}(1, 5)$



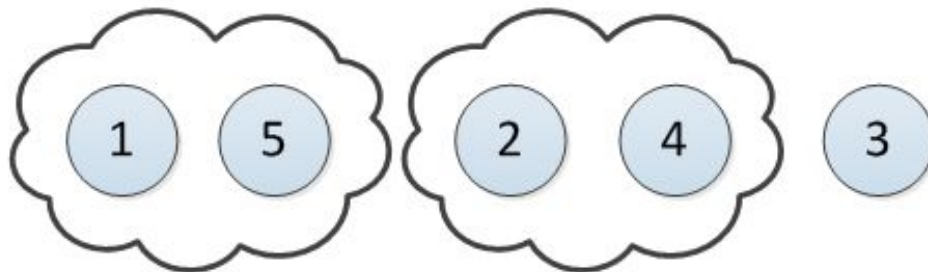
# Система Непересекающихся Множеств - CHM

Disjoint Set



$\text{find}(2) = 2$

$\text{union}(1, 5)$   
 $\text{union}(2, 4)$



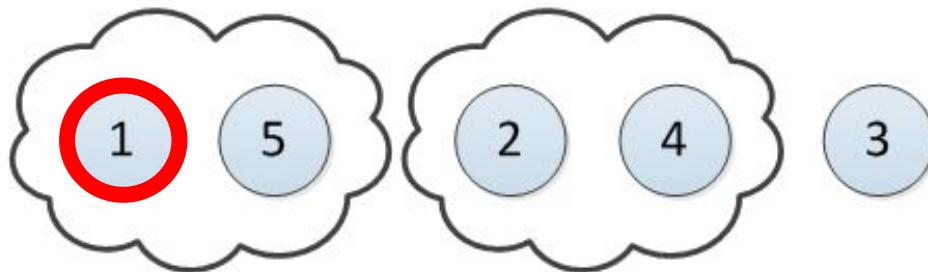
# Система Непересекающихся Множеств - CHM

Disjoint Set



$\text{find}(2) = 2$

$\text{union}(1, 5)$   
 $\text{union}(2, 4)$



$\text{find}(1) = 1$

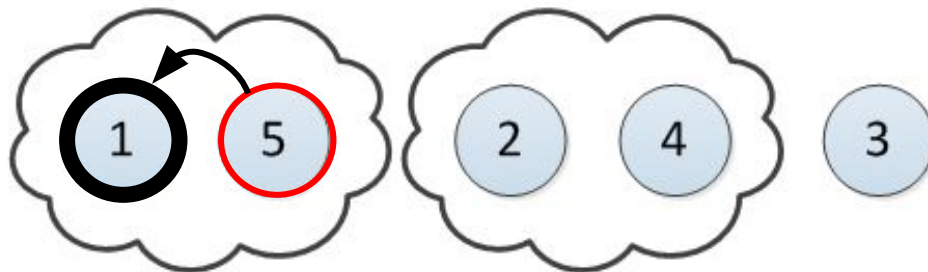
# Система Непересекающихся Множеств - CHM

Disjoint Set



$\text{find}(2) = 2$

$\text{union}(1, 5)$   
 $\text{union}(2, 4)$



$\text{find}(1) = 1$

$\text{find}(5) = 1$

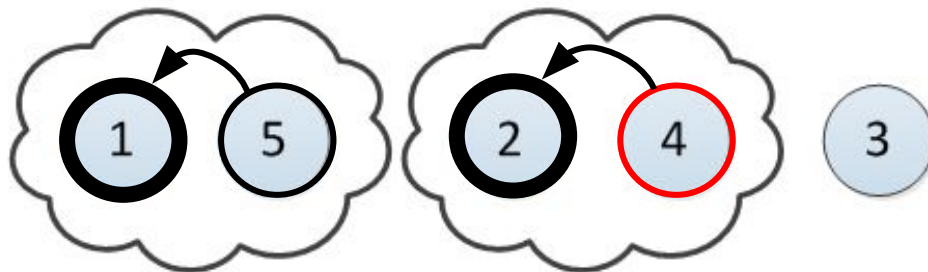
# Система Непересекающихся Множеств - CHM

Disjoint Set



$\text{find}(2) = 2$

$\text{union}(1, 5)$   
 $\text{union}(2, 4)$



$\text{find}(1) = 1$

$\text{find}(5) = 1$

$\text{find}(4) = 2$

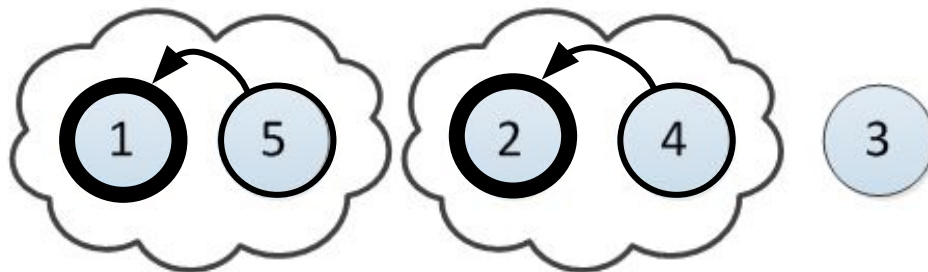
# Система Непересекающихся Множеств - CHM

Disjoint Set



$\text{find}(2) = 2$

$\text{union}(1, 5)$   
 $\text{union}(2, 4)$

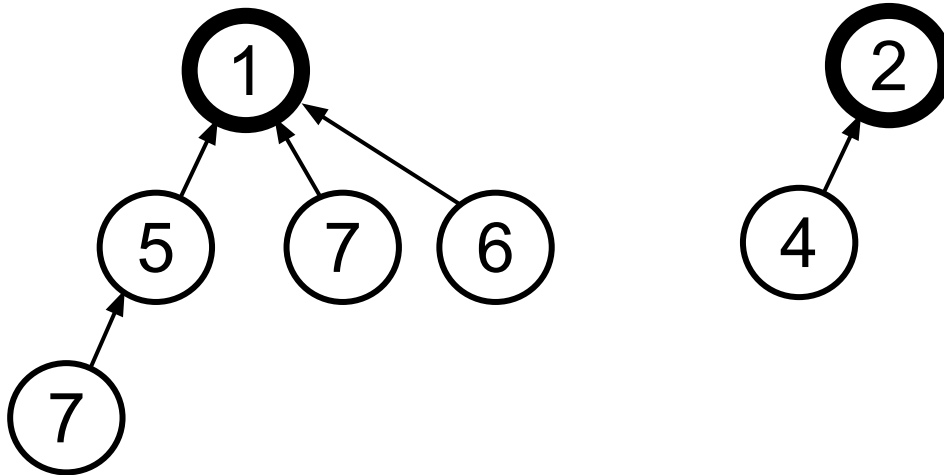


$\text{find}(1) = 1$   
 $\text{find}(5) = 1$   
 $\text{find}(4) = 2$

Как сделать так чтобы любая операция работала за  **$O(1)$** ? (амортизировано)

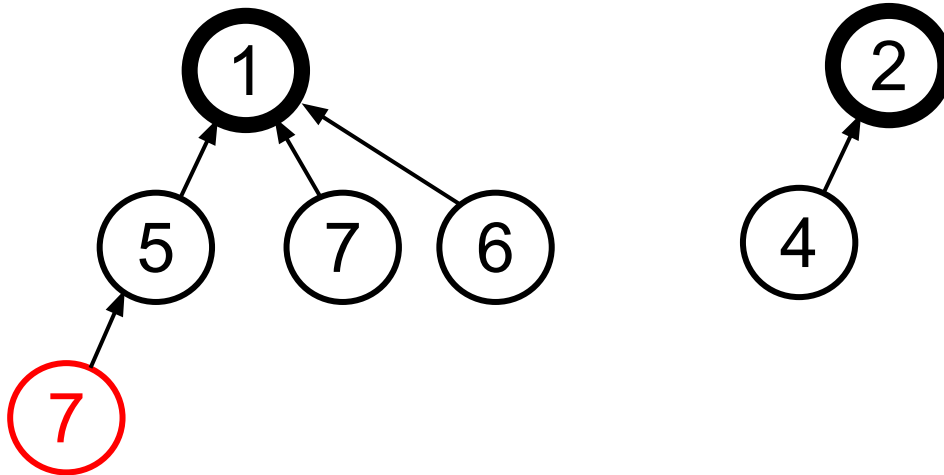
# Система Непересекающихся Множеств - CHM

## Disjoint Set



# Система Непересекающихся Множеств - CHM

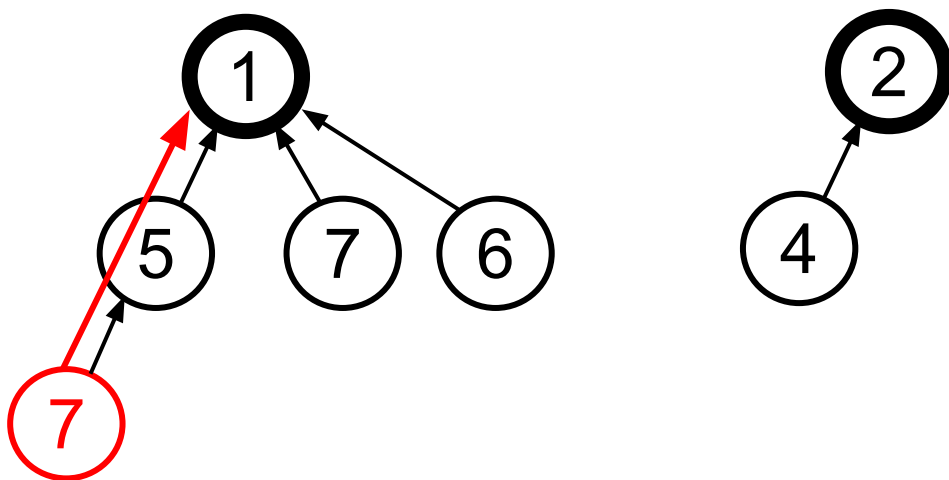
## Disjoint Set



get(7) **как выполнить?**

# Система Непересекающихся Множеств - CHM

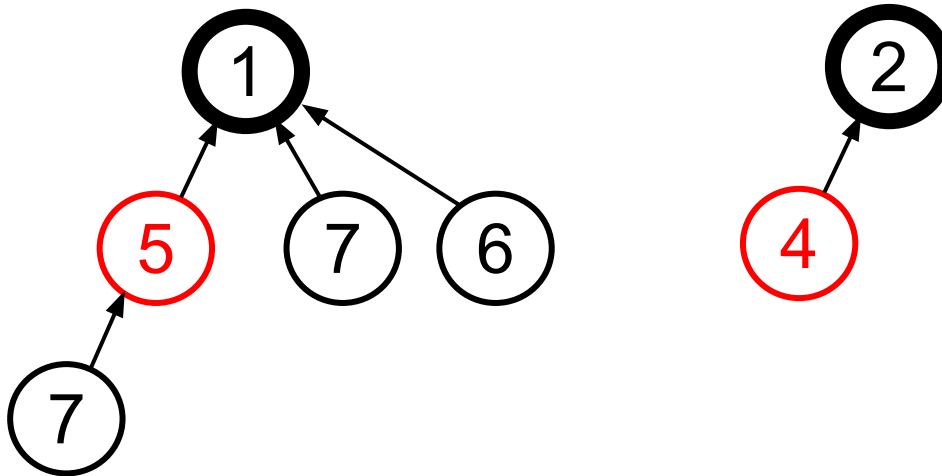
## Disjoint Set



get(7) **как выполнить?**

# Система Непересекающихся Множеств - CHM

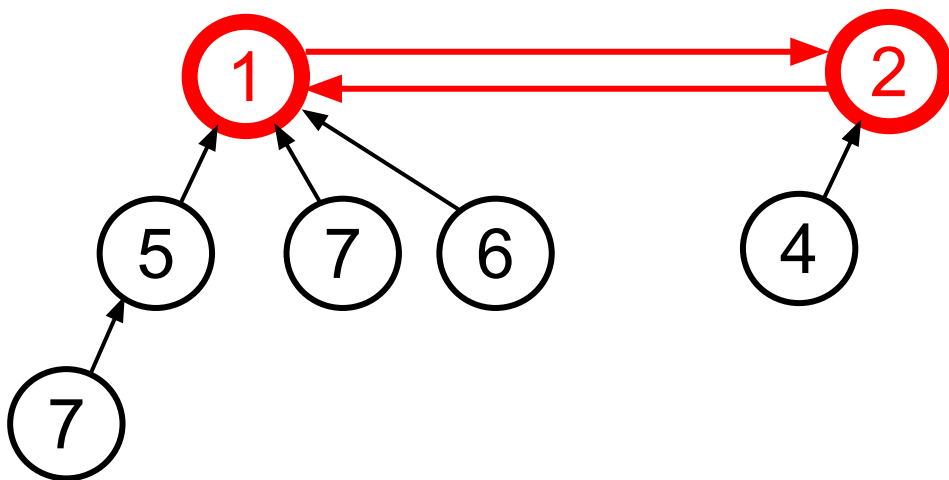
## Disjoint Set



`union(5, 4)` как выполнить?

# Система Непересекающихся Множеств - CHM

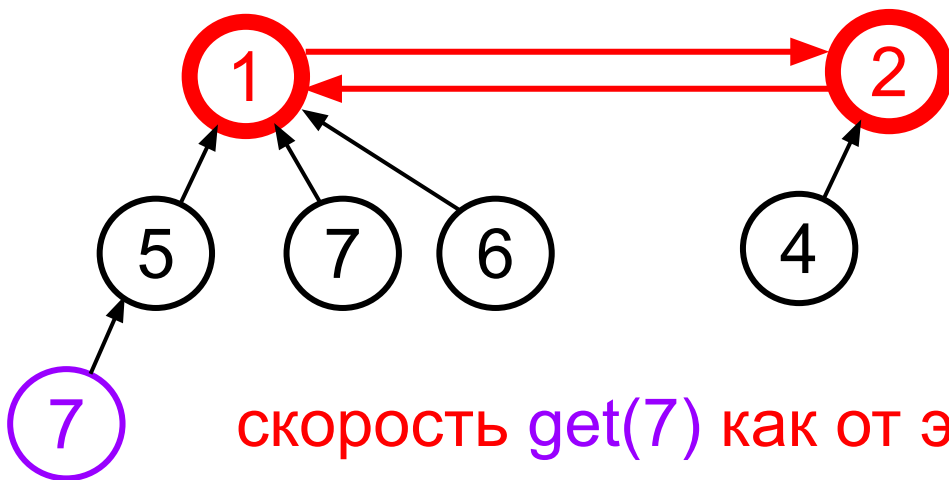
## Disjoint Set



`union(1, 2)` как выполнить?

# Система Непересекающихся Множеств - CHM

## Disjoint Set

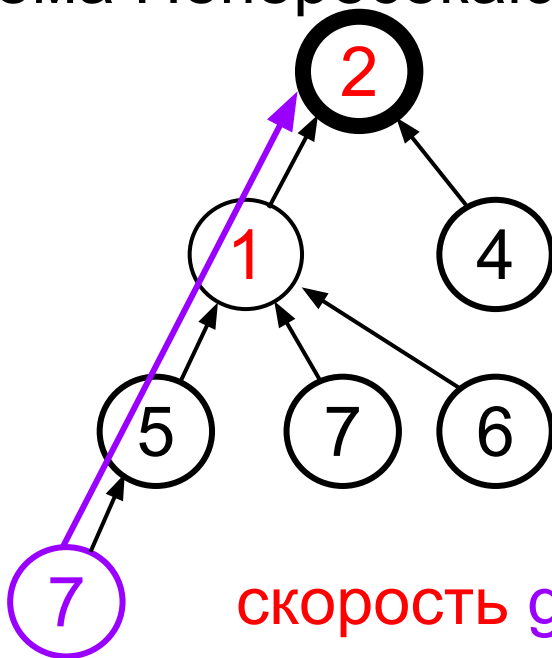


скорость `get(7)` как от этого зависит?

`union(1, 2)` как выполнить?

# Система Непересекающихся Множеств - CHM

Disjoint Set

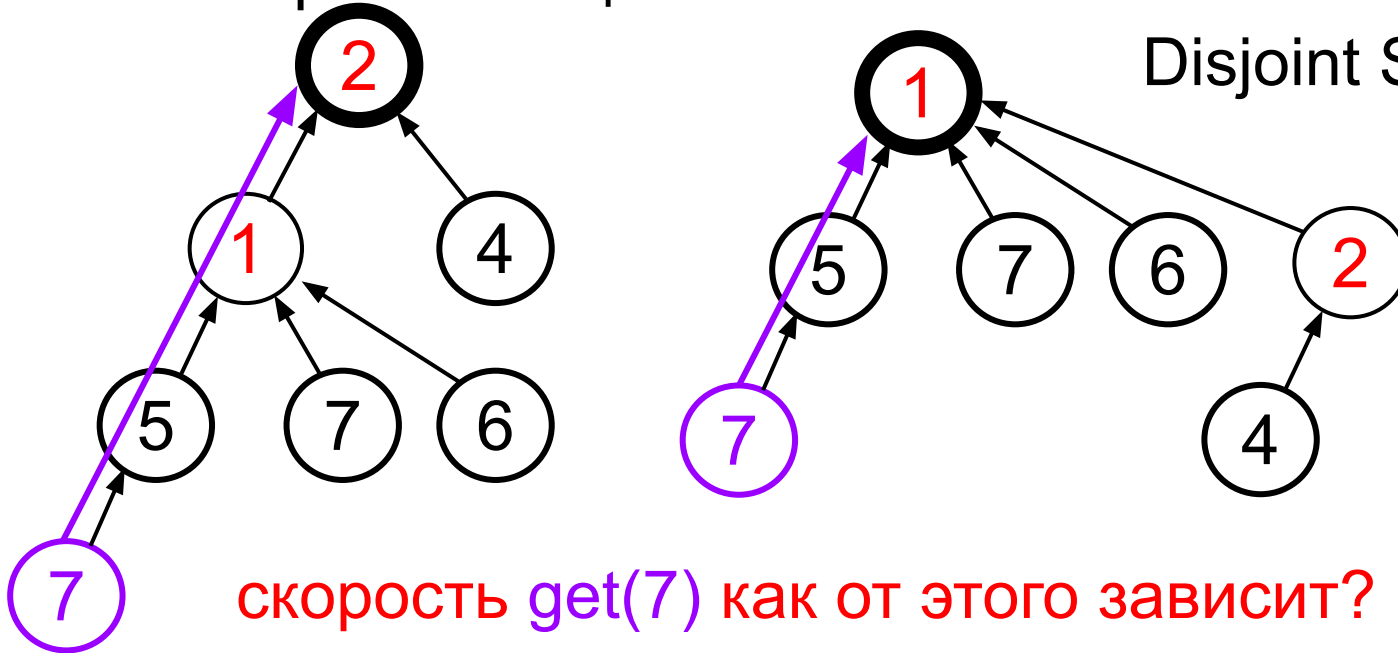


скорость `get(7)` как от этого зависит?

`union(1, 2)` как выполнить?

# Система Непересекающихся Множеств - CHM

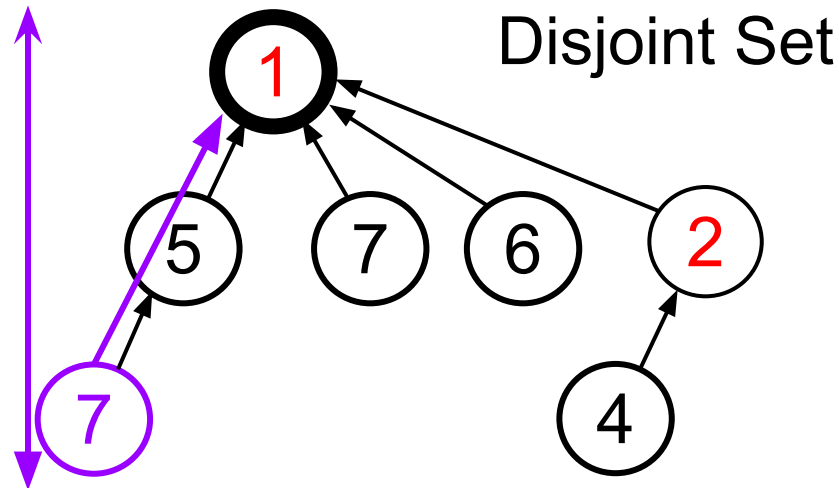
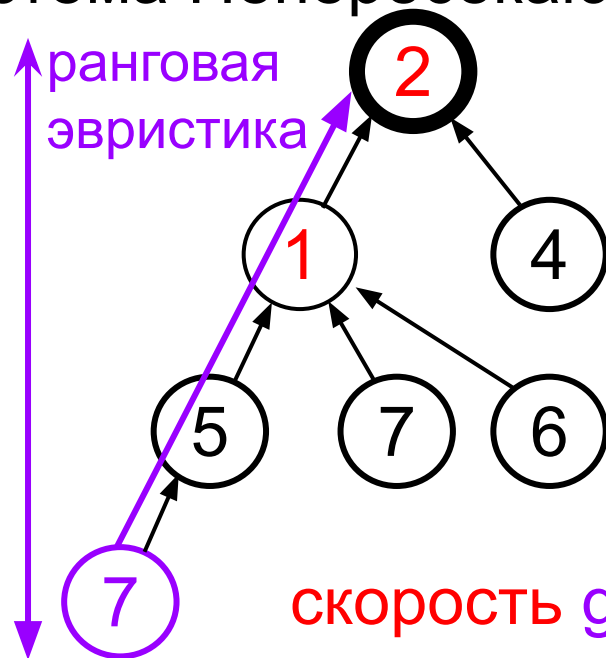
Disjoint Set



скорость `get(7)` как от этого зависит?

`union(1, 2)` как выполнить?

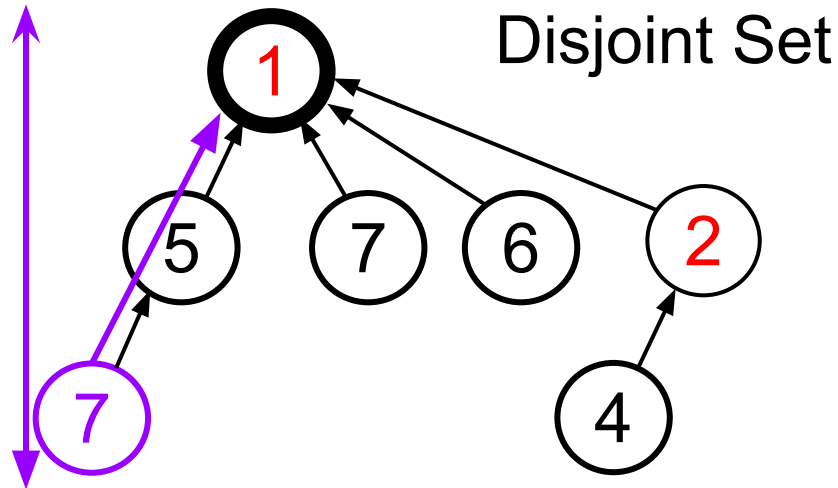
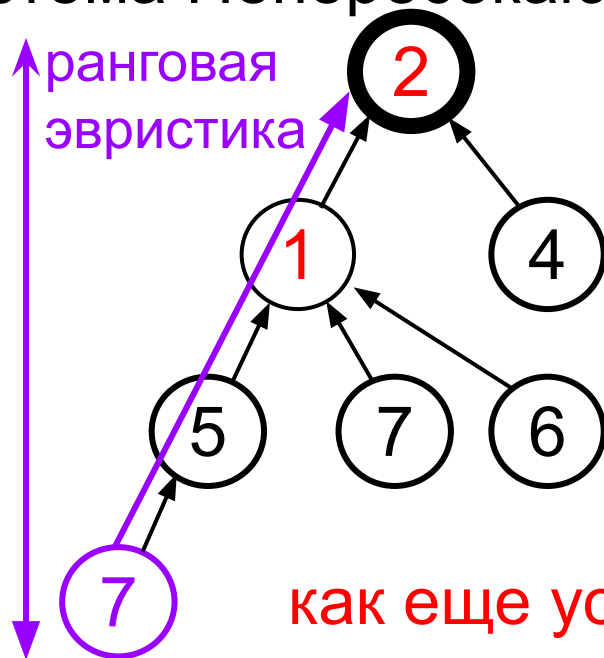
# Система Непересекающихся Множеств - CHM



скорость `get(7)` как от этого зависит?

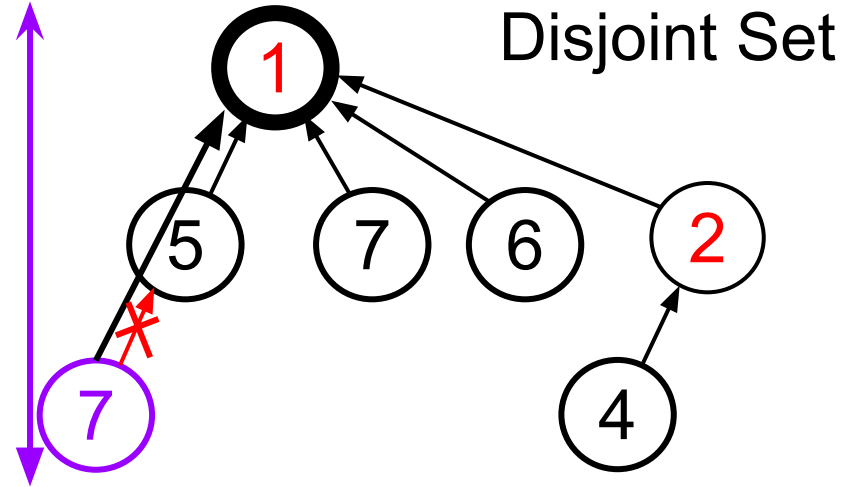
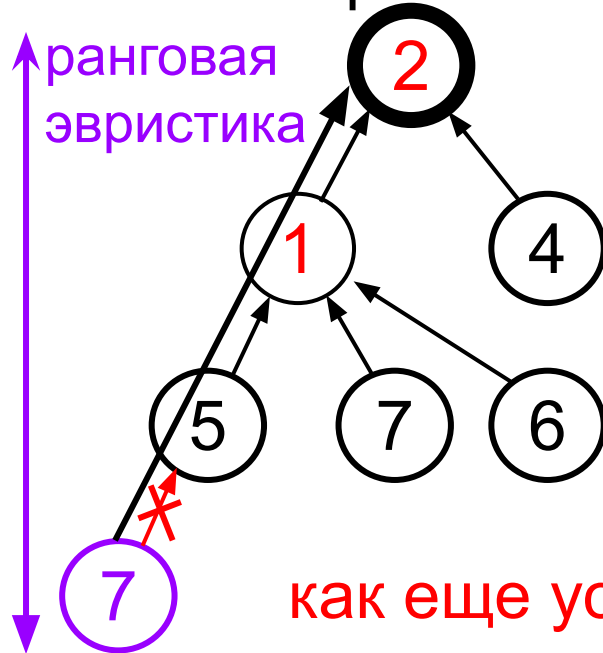
`union(1, 2)` как выполнить?

# Система Непересекающихся Множеств - CHM

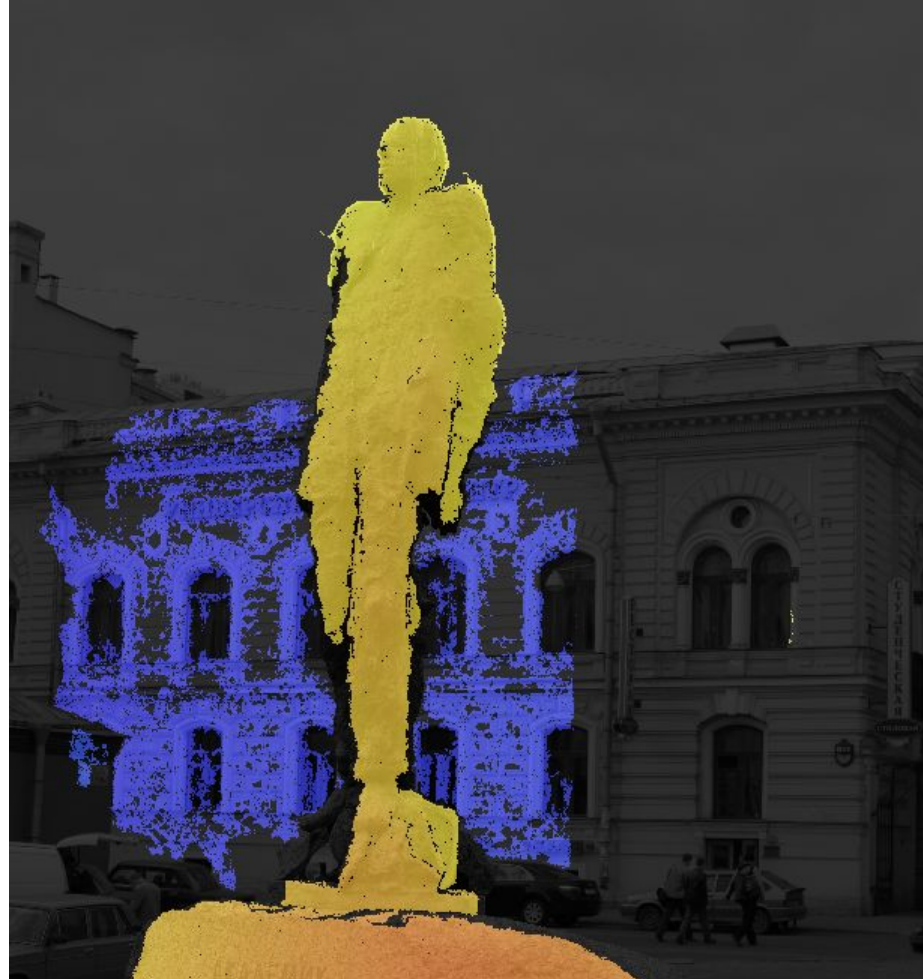
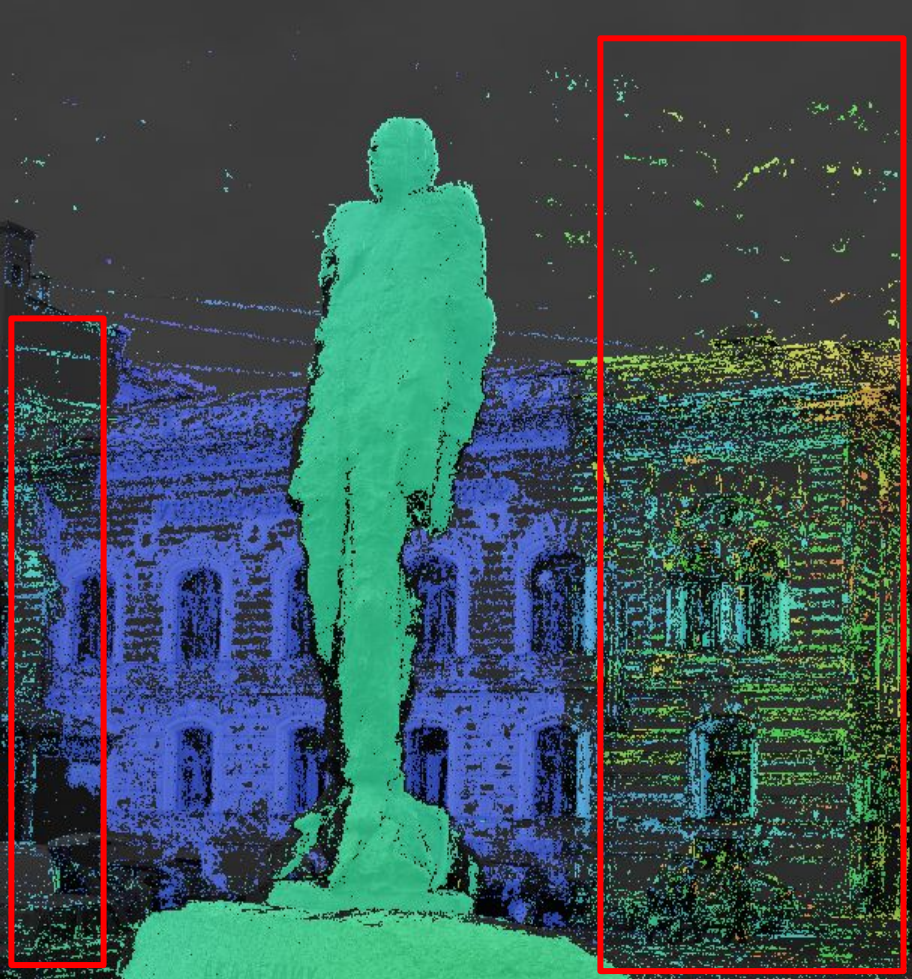


как еще ускорить `get(7)` если он частый?

# Система Непересекающихся Множеств - CHM

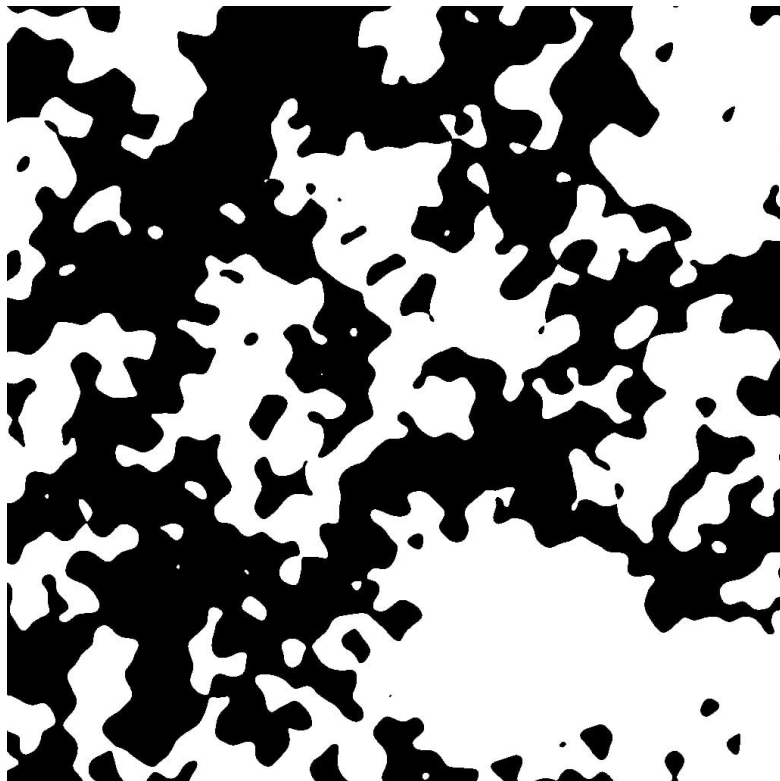


как еще ускорить `get(7)` если он частый?  
сжатие путей!



# Система Непересекающихся Множеств - CHM

## Disjoint Set

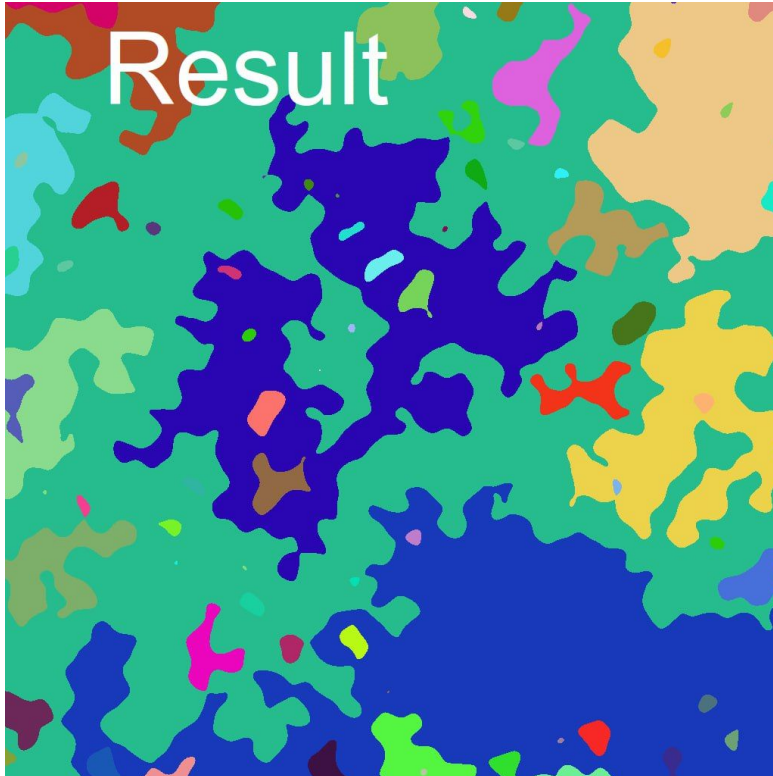


**Как найти площадь каждого острова?**

(например для speckles filtering -  
фильтрации выбросов)

# Система Непересекающихся Множеств - CHM

## Disjoint Set

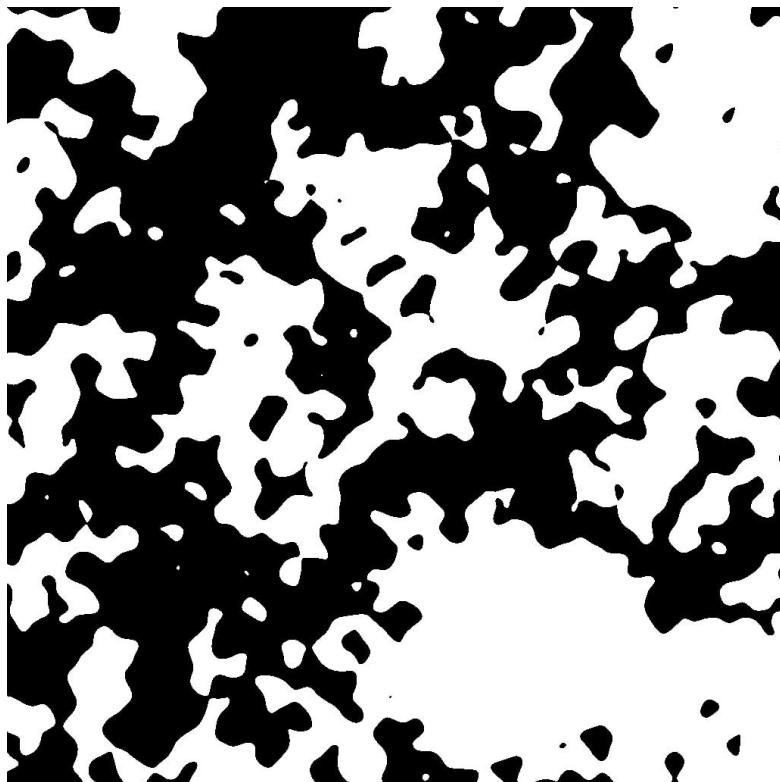


**Как найти площадь каждого острова?**

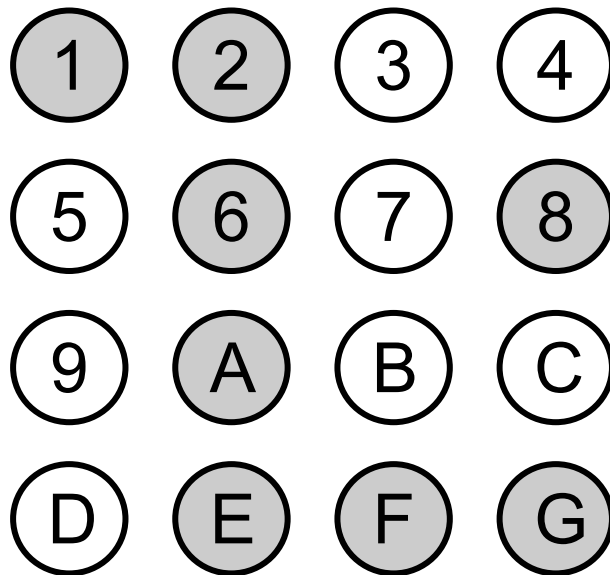
(например для speckles filtering -  
фильтрации выбросов)

# Система Непересекающихся Множеств - CHM

Disjoint Set

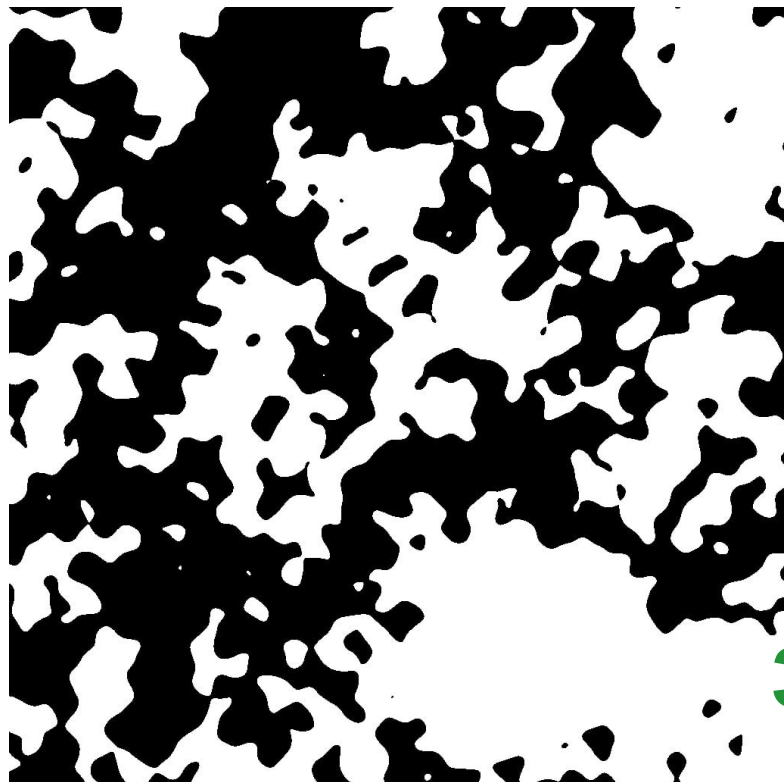


Как найти площадь каждого острова?

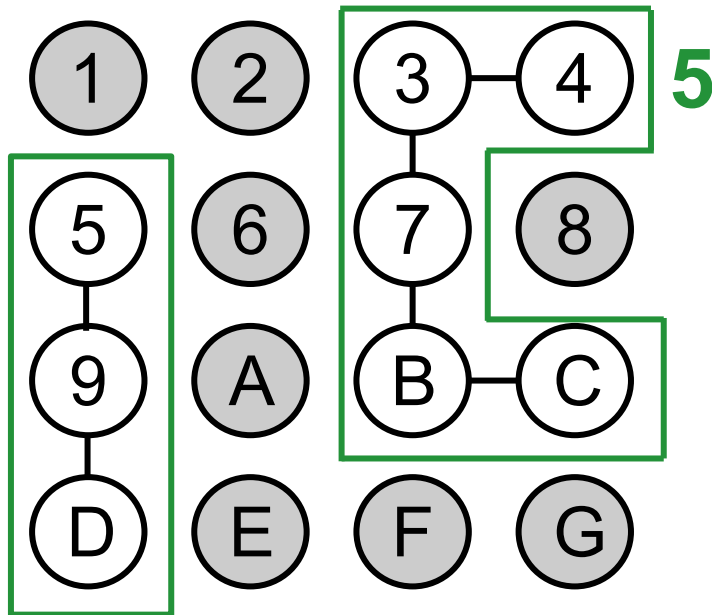


# Система Непересекающихся Множеств - CHM

Disjoint Set

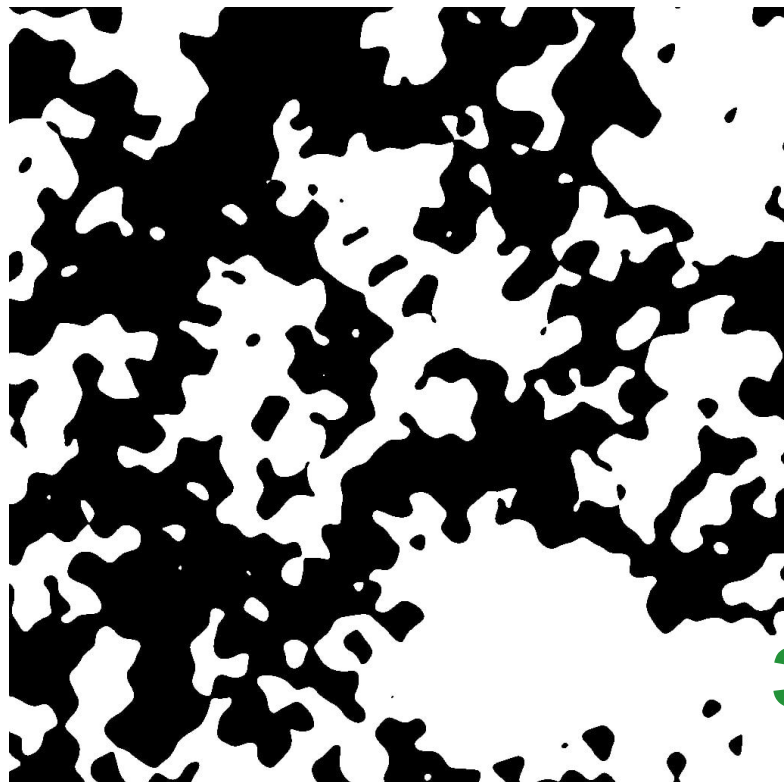


Как найти площадь каждого острова?

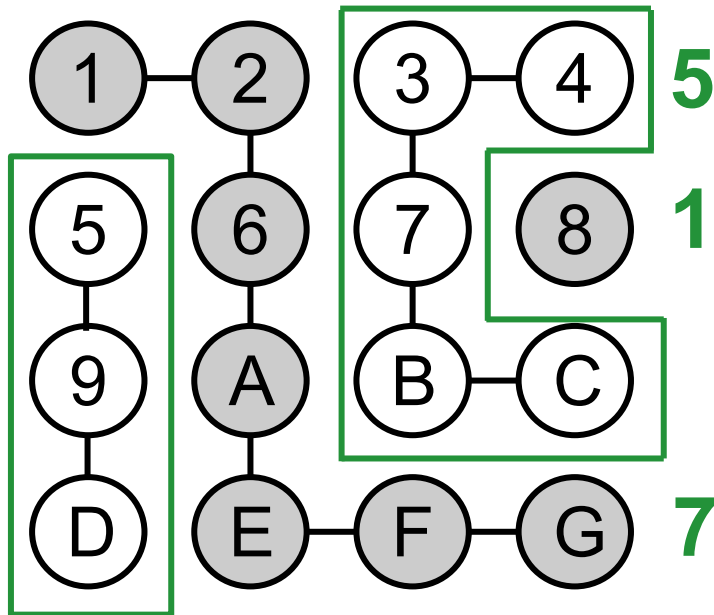


# Система Непересекающихся Множеств - CHM

Disjoint Set

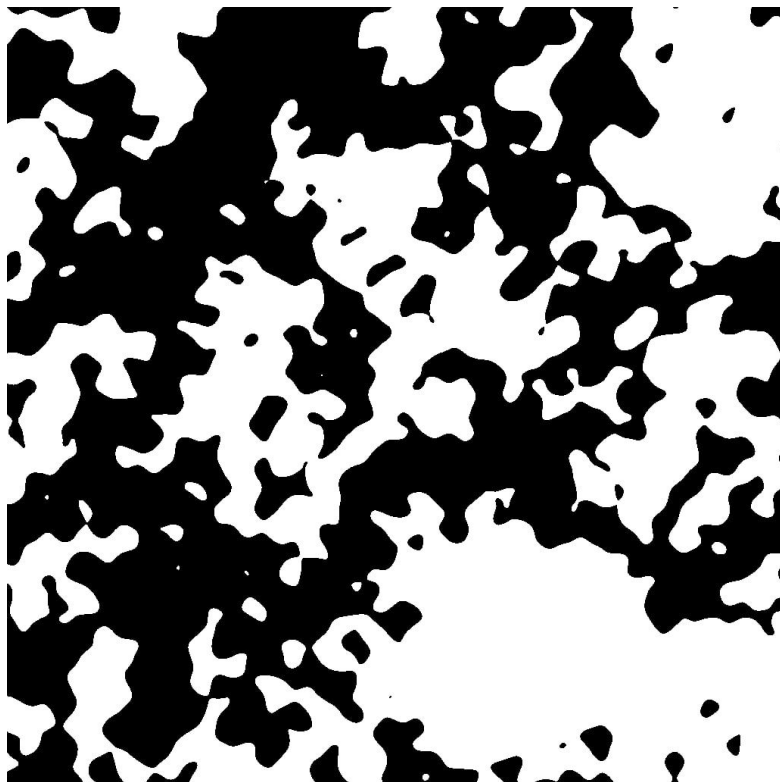


Как найти площадь каждого острова?

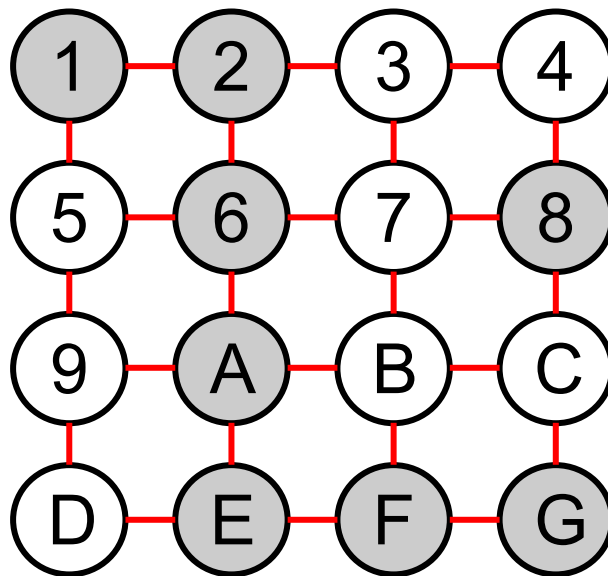


# Система Непересекающихся Множеств - CHM

## Disjoint Set



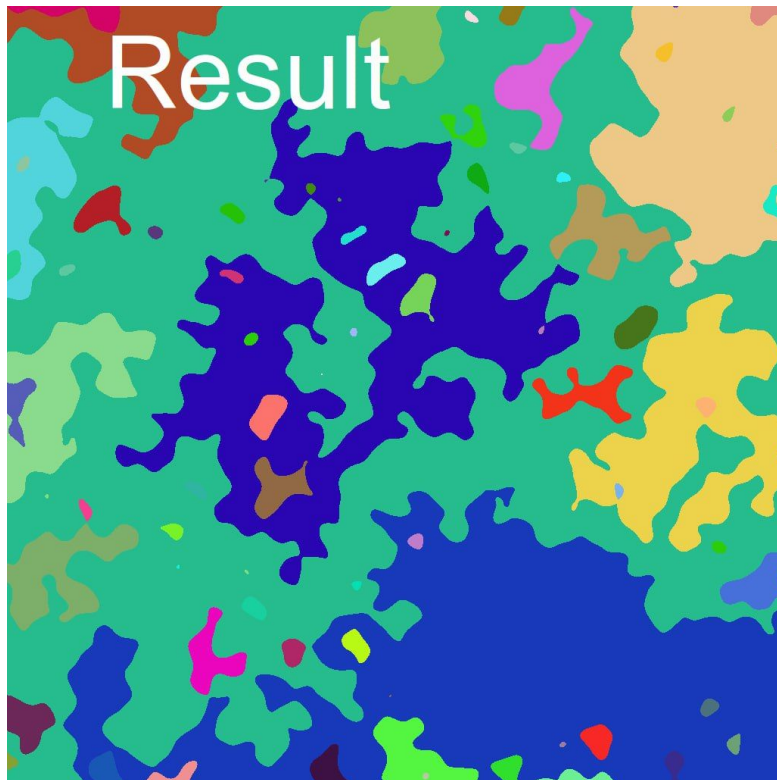
Как найти площадь каждого острова?



Можем ли  
выполнять все  
**union** операции  
параллельно?

# Система Непересекающихся Множеств - CHM

## Disjoint Set



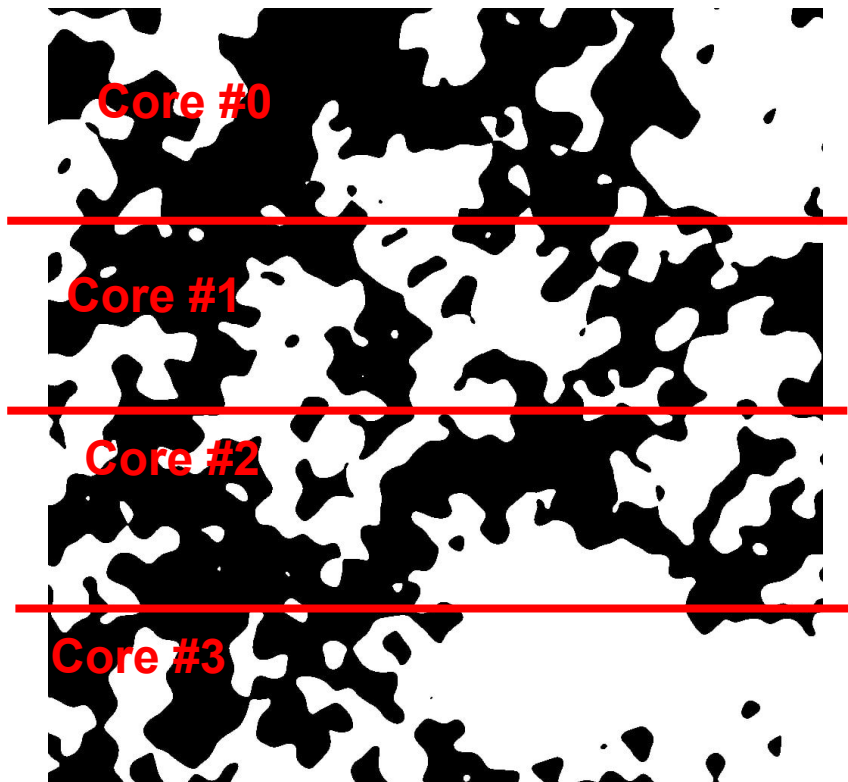
Как найти площадь каждого острова?

(например для speckles filtering -  
фильтрации выбросов)

**Как распараллелить на CPU?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



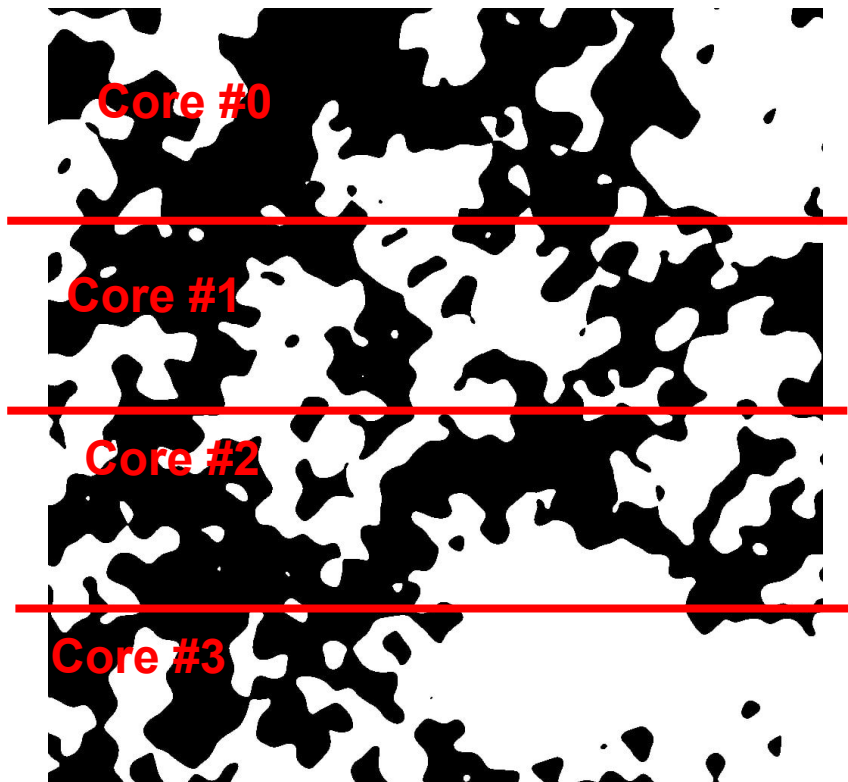
Как найти площадь каждого острова?

(например для speckles filtering -  
фильтрации выбросов)

**Как распараллелить на CPU?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

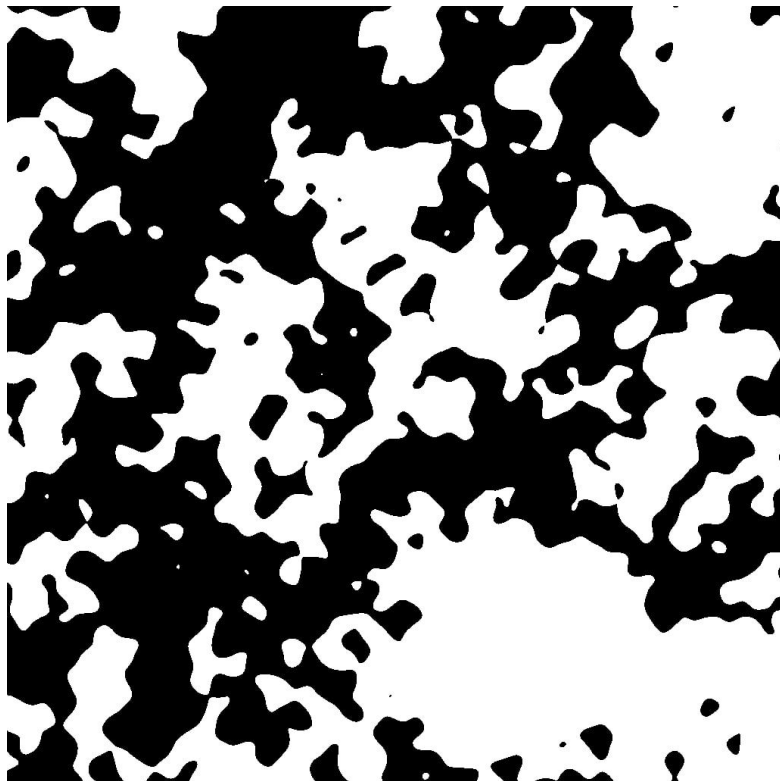
(например для speckles filtering -  
фильтрации выбросов)

**Как распараллелить на CPU?**

**А как потом доделать работу на швах?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

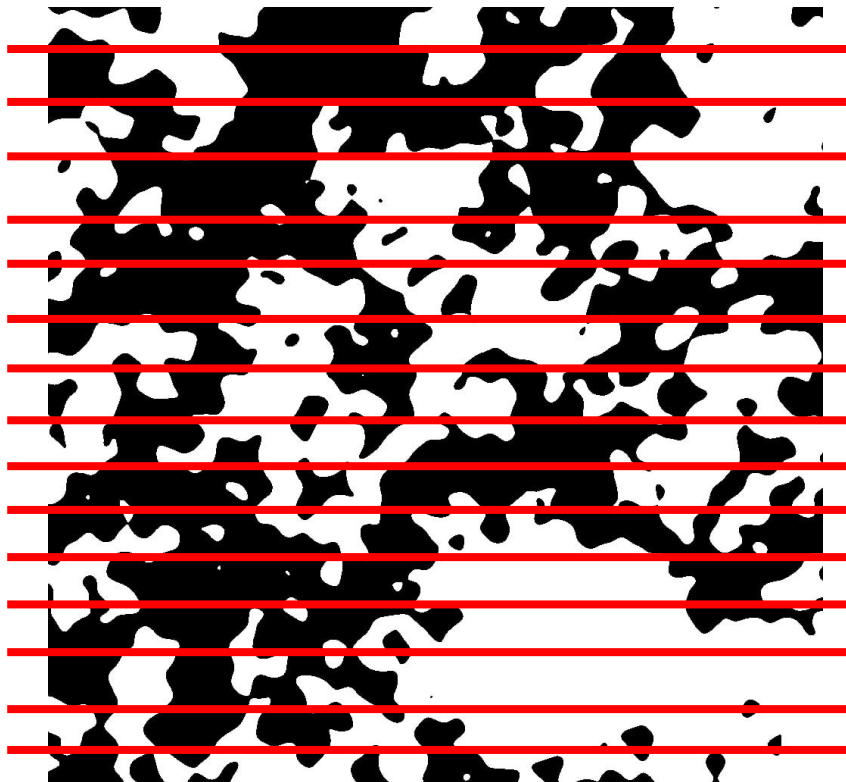
(например для speckles filtering -  
фильтрации выбросов)

Как распараллелить на CPU?

**Как распараллелить на GPU?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

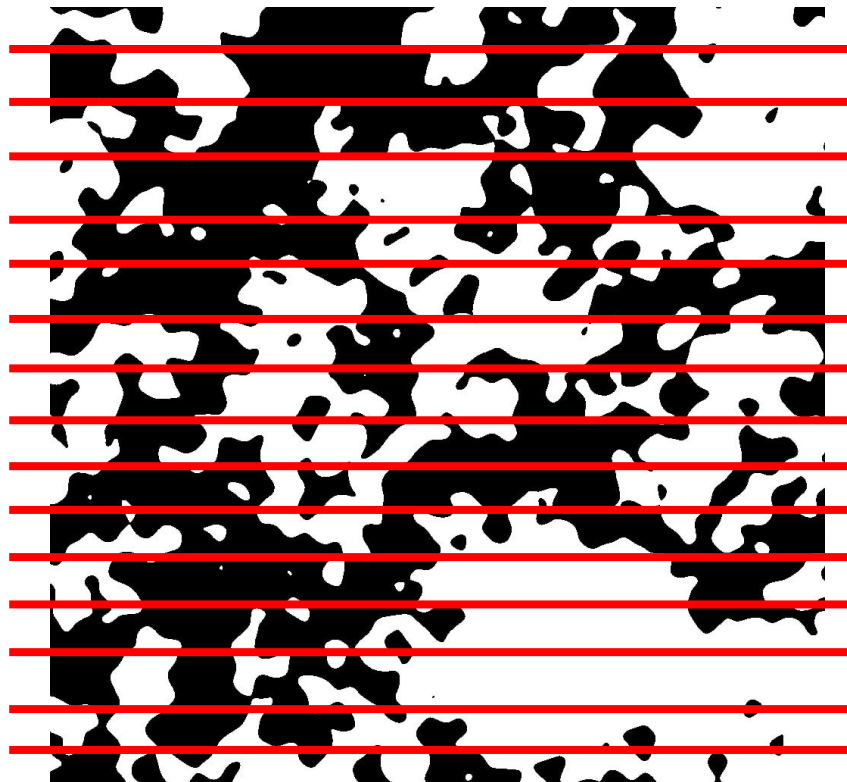
(например для speckles filtering -  
фильтрации выбросов)

Как распараллелить на CPU?

**Как распараллелить на GPU?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

(например для speckles filtering -  
фильтрации выбросов)

Как распараллелить на CPU?

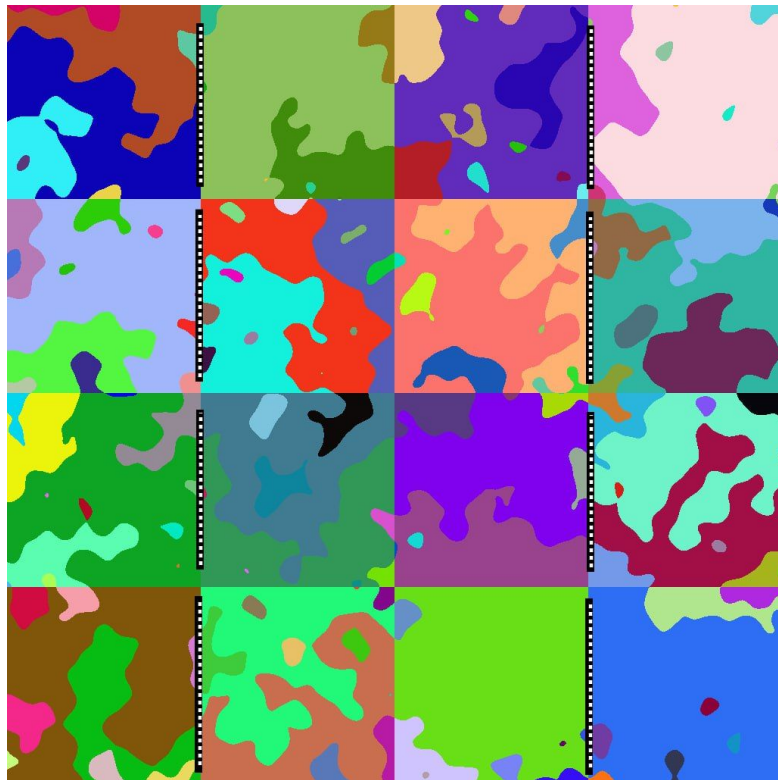
**Как распараллелить на GPU?**

**А как потом доделать работу на швах?**

Проблема как с последним этапом merge,  
но тут **совсем кошмар!**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

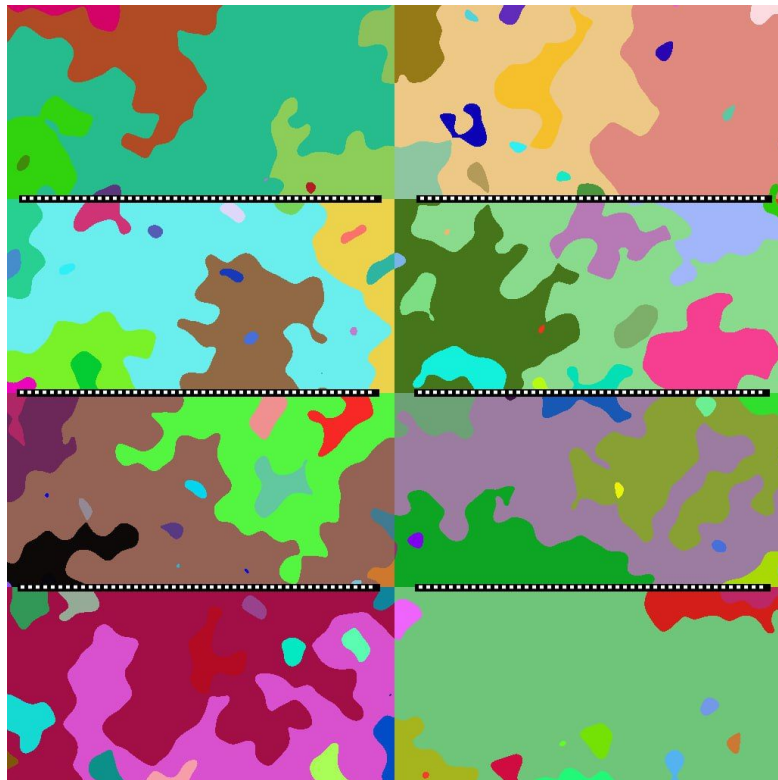
(например для speckles filtering -  
фильтрации выбросов)

Как распараллелить на CPU?

**Как распараллелить на GPU?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

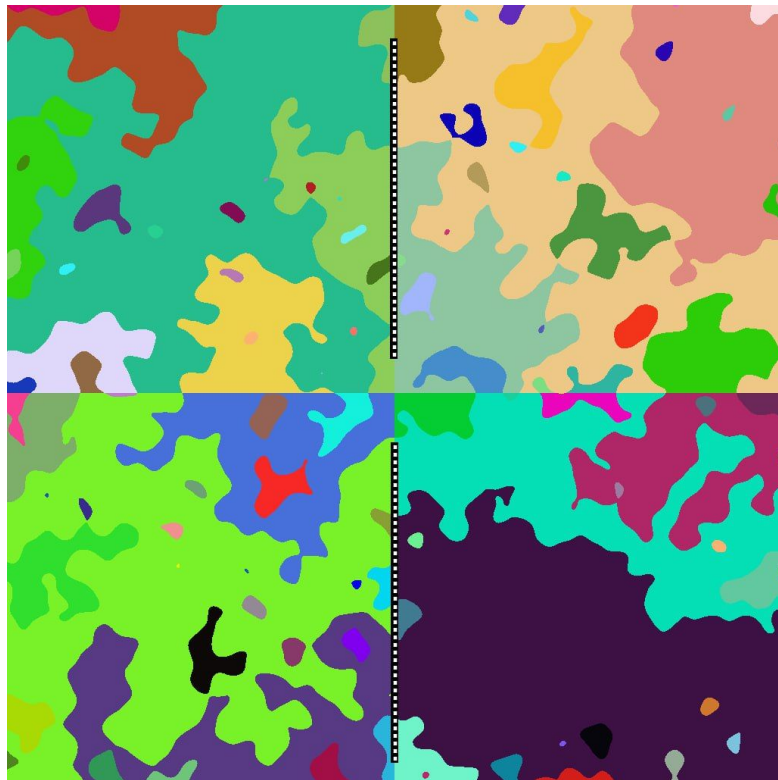
(например для speckles filtering -  
фильтрации выбросов)

Как распараллелить на CPU?

**Как распараллелить на GPU?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

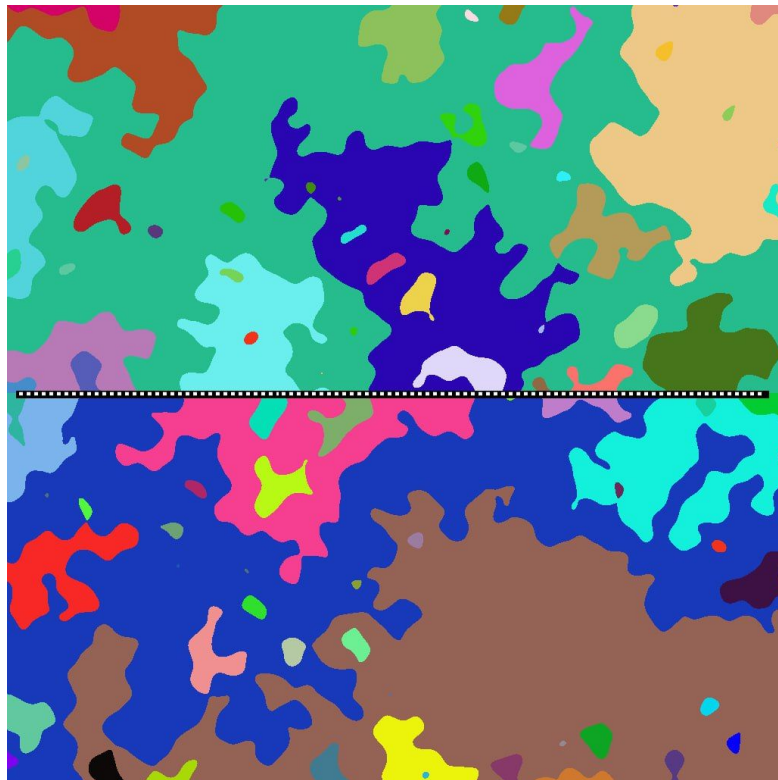
(например для speckles filtering -  
фильтрации выбросов)

Как распараллелить на CPU?

**Как распараллелить на GPU?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

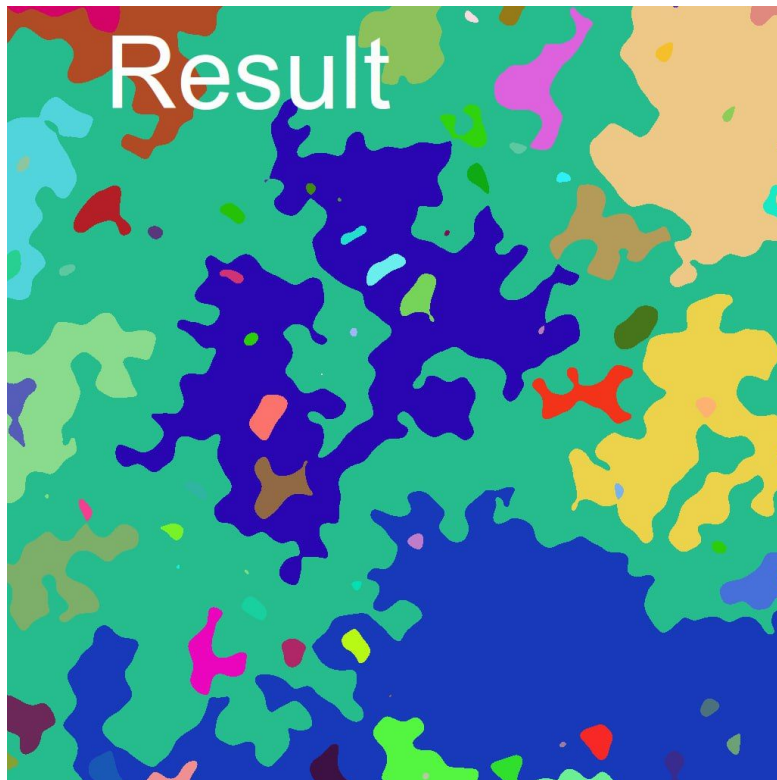
(например для speckles filtering -  
фильтрации выбросов)

Как распараллелить на CPU?

**Как распараллелить на GPU?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



Как найти площадь каждого острова?

(например для speckles filtering -  
фильтрации выбросов)

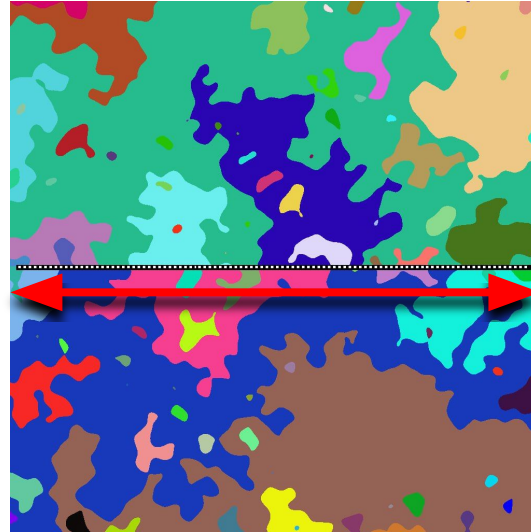
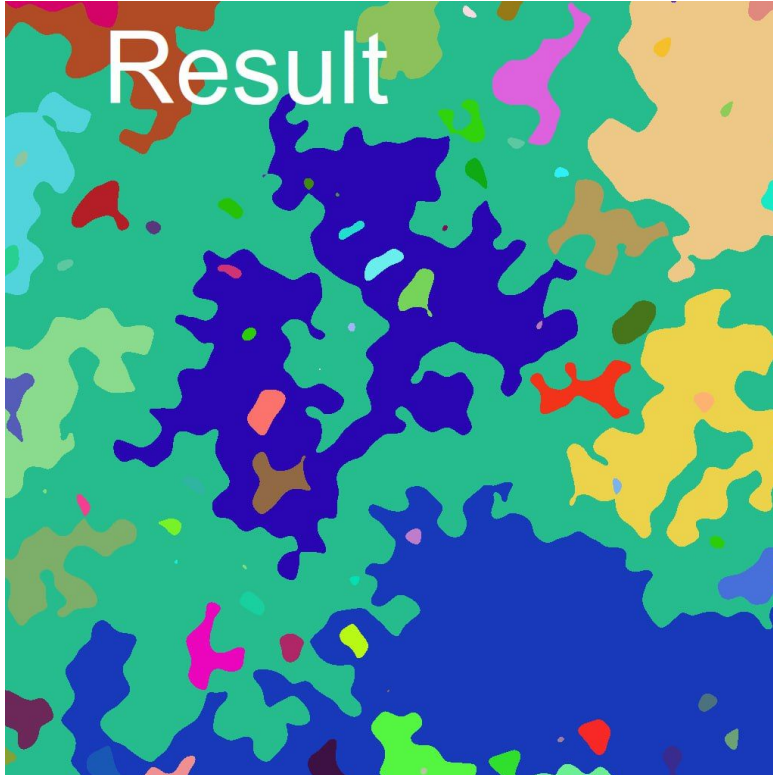
Как распараллелить на CPU?

Как распараллелить на GPU?

**Какой из запусков ядра самый  
медленный?**

# Система Непересекающихся Множеств - CHM

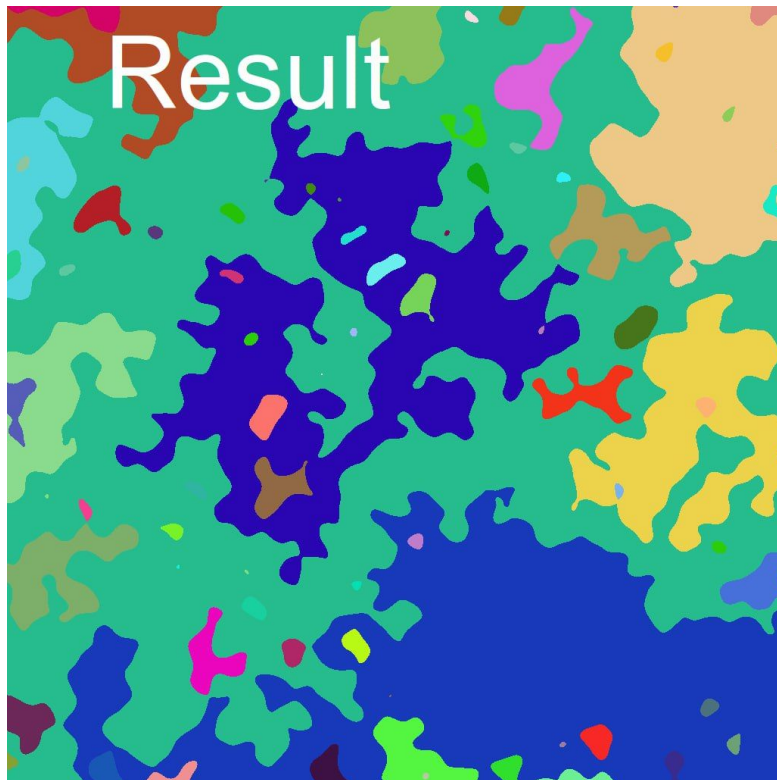
Disjoint Set



**Какой из запусков ядра самый медленный?**

# Система Непересекающихся Множеств - CHM

## Disjoint Set



|                      |                  |
|----------------------|------------------|
| Картинка:            | <b>4096x4096</b> |
| Множеств:            | <b>91</b>        |
| CPU 5960X:           | <b>2.8 sec</b>   |
| CPU 5960X OpenMP:    | <b>0.73 sec</b>  |
| GPU RTX 3080 Vulkan: | <b>0.094 sec</b> |



CS Space

Клуб технологий и науки

Вопросы?



[@UnicornGlade](https://t.me/UnicornGlade)



[@PolarNick239](https://t.me/PolarNick239)



[polarnick239@gmail.com](mailto:polarnick239@gmail.com)



Николай Полярный

Activate Ubuntu

Go to Settings to activate Ubuntu.