



Вычисления на видеокартах

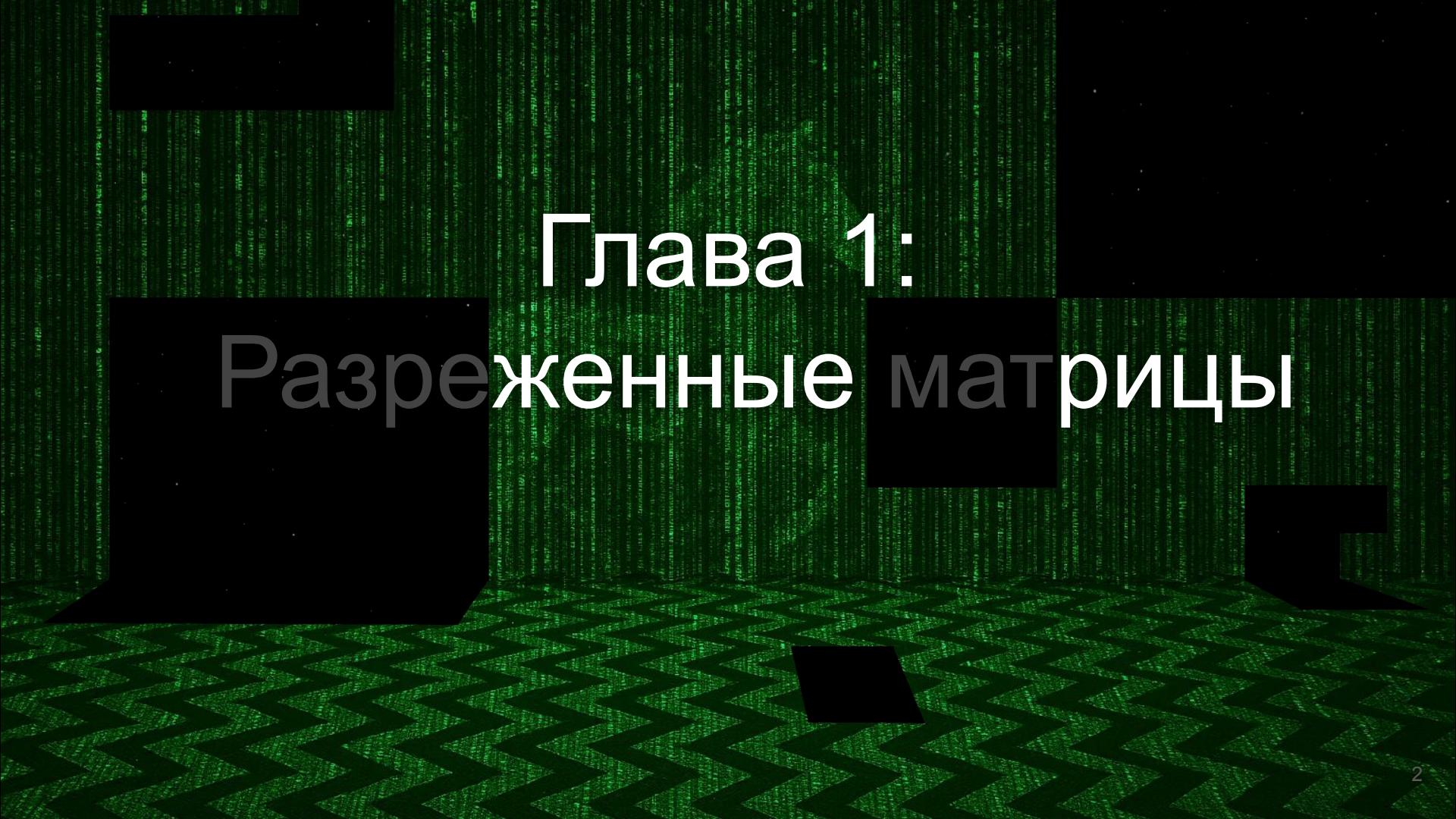


Лекция 8 - Разреженные матрицы

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

(Системы Непересекающихся Множеств)

Глава 1: Разреженные матрицы





SPbu



NVIDIA
TESLA



CUDIA

1

SPbu

OpenCL

2

Chernic.

OpenCL

3

SPbu

OpenCL

OpenCL



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 $N \times M$? 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 $N \times M$? 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 $N \times M$? 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*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

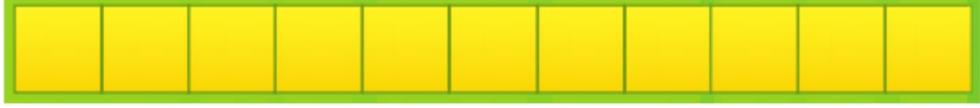
	0	1	2	3	4
0	1	2		11	
1		3	4		
2		5	6	7	
3				8	
4				9	10

Что **САМОЕ** простое можно придумать?

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:	

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

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 2

col:

0 2

val:

1 | 2

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) 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	?							
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) 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	...					
col:	0	1	3	1	2	...					
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

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) COOrdinate-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) COOrdinate-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

Насколько все хорошо?

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

1) COOrdinate-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

Итеративные методы:

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

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

Насколько все хорошо?

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

1) COOrdinate-wise - COO

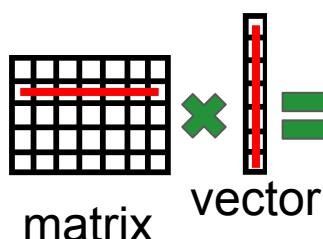
row:	0	0	0	1	1	2	2	2	3	4	4
col:	0	1	3	1	2	1	2	3	3	3	4
val:	1	2	1	3	4	5	6	7	8	9	10

Итеративные методы:

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

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

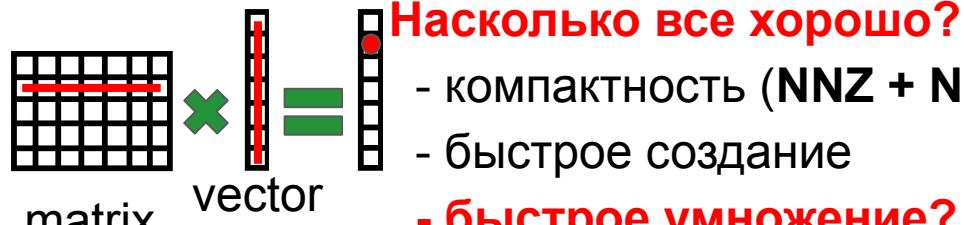
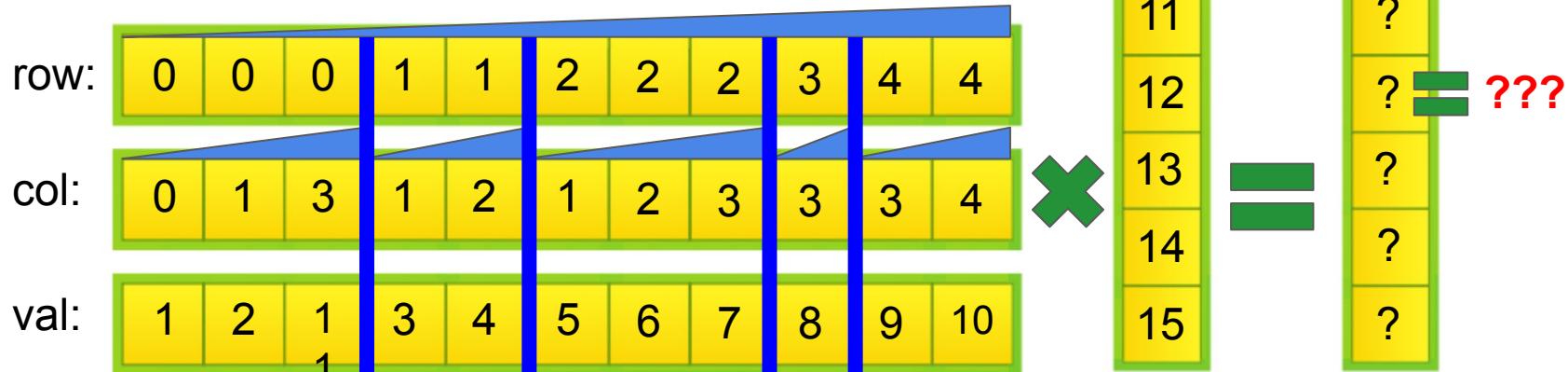
SpMV (Sparse matrix–vector multiplication)



Насколько все хорошо?

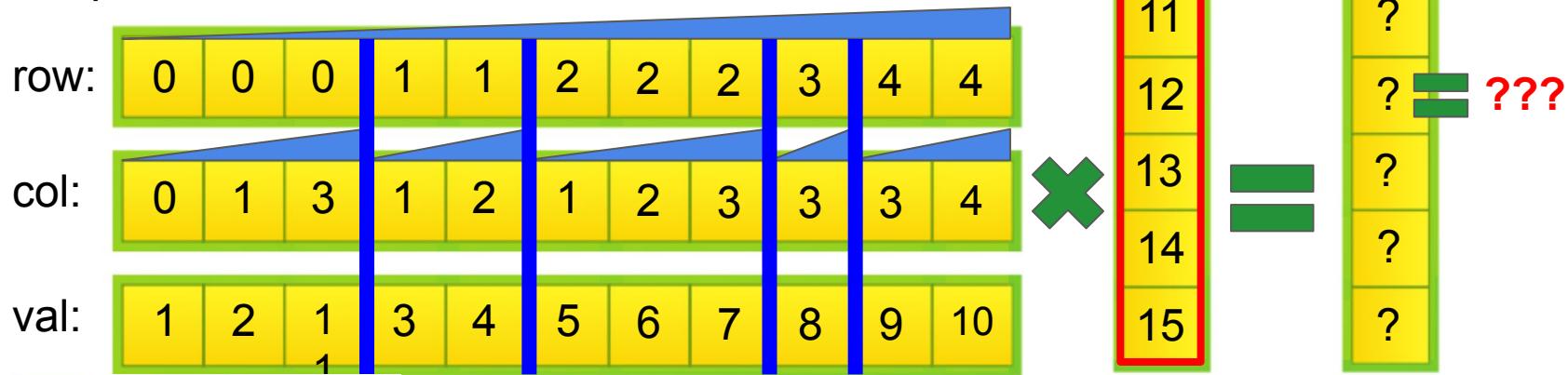
- компактность ($NNZ + NNZ + NNZ$)
- быстрое создание
- быстрое умножение?

1) COOrdinate-wise - COO

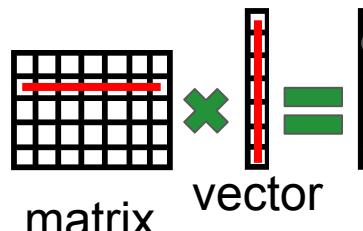


Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

1) COOrdinate-wise - COO



SpMV (Sparse matrix–vector multiplication)

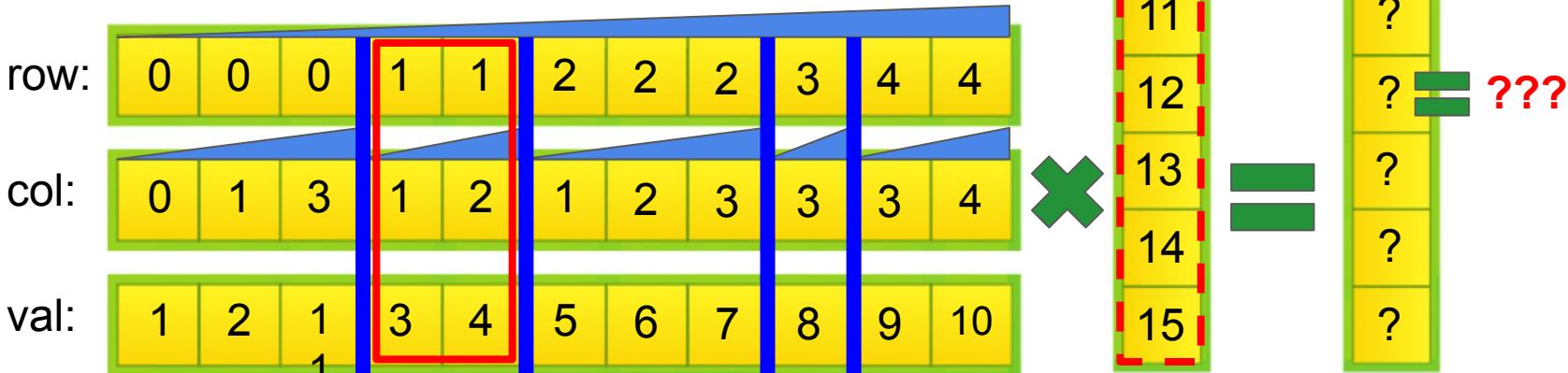


Насколько все хорошо?

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

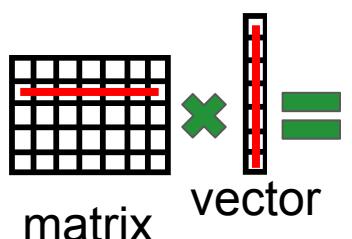
1) COOrdinate-wise - COO

Умножаем на
какие элементы?



1	2		11	
		3	4	
		5	6	7
			8	
			9	10

SpMV (Sparse matrix–vector multiplication)



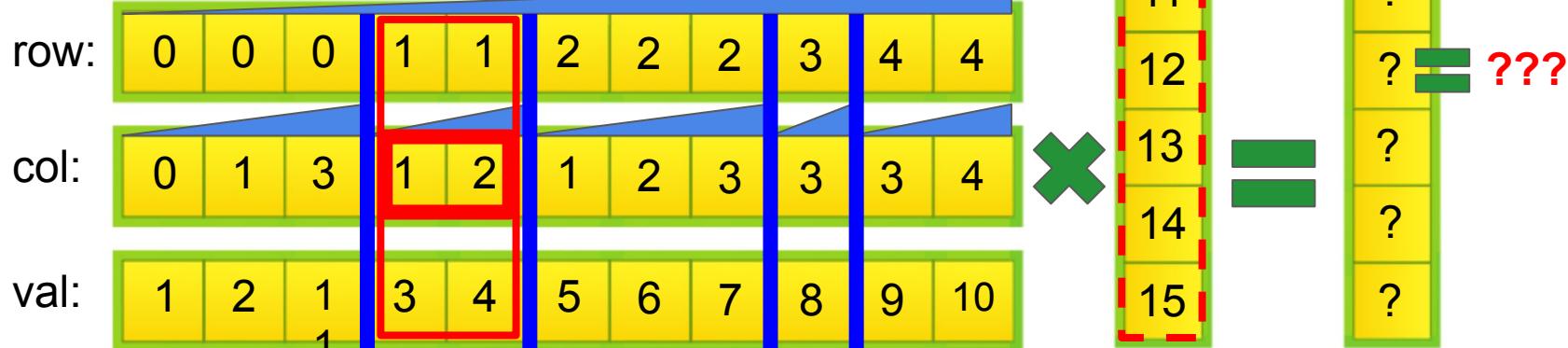
Насколько все хорошо?

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

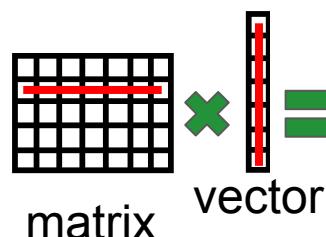
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

1) COOrdinate-wise - COO

Умножаем на
какие элементы?



SpMV (Sparse matrix–vector multiplication)



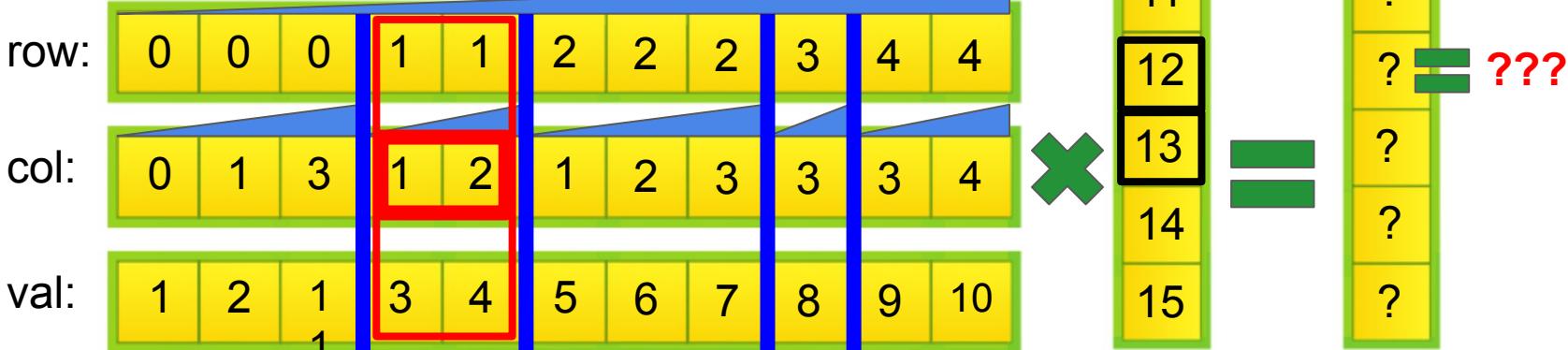
Насколько все хорошо?

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

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

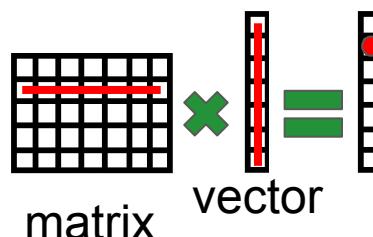
Умножаем на
какие элементы?

1) COOrdinate-wise - COO



1	2		11
3	4		
5	6	7	
			8
			9 10

SpMV (Sparse matrix–vector multiplication)

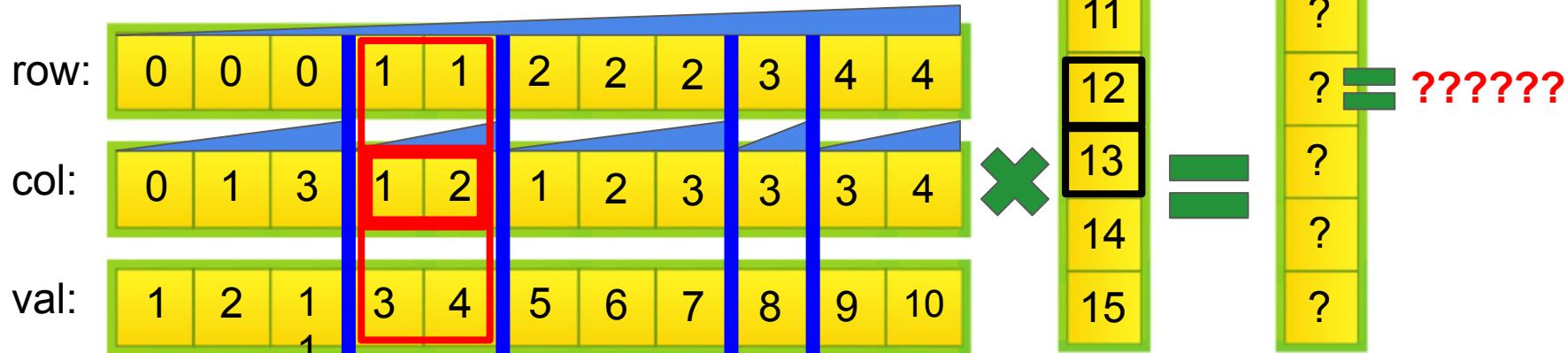


Насколько все хорошо?

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

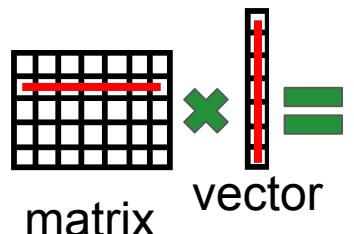
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

1) COOrdinate-wise - COO



1	2		11
3	4		
5	6	7	
			8
			9 10

SpMV (Sparse matrix–vector multiplication)

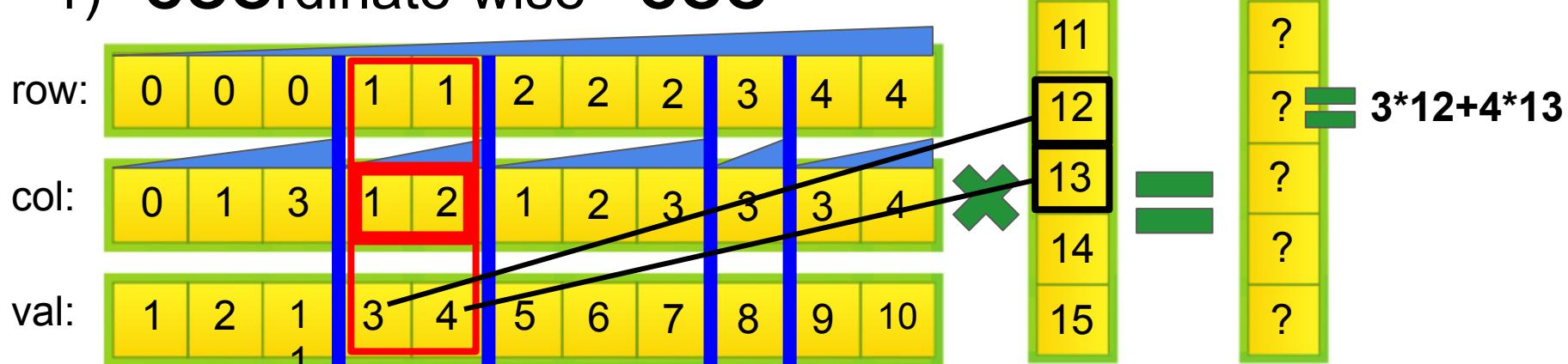


Насколько все хорошо?

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

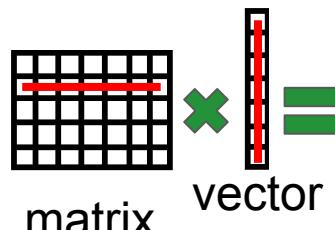
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

1) COOrdinate-wise - COO



1	2		11
		3 4	
		5 6	7
			8
			9 10

SpMV (Sparse matrix–vector multiplication)

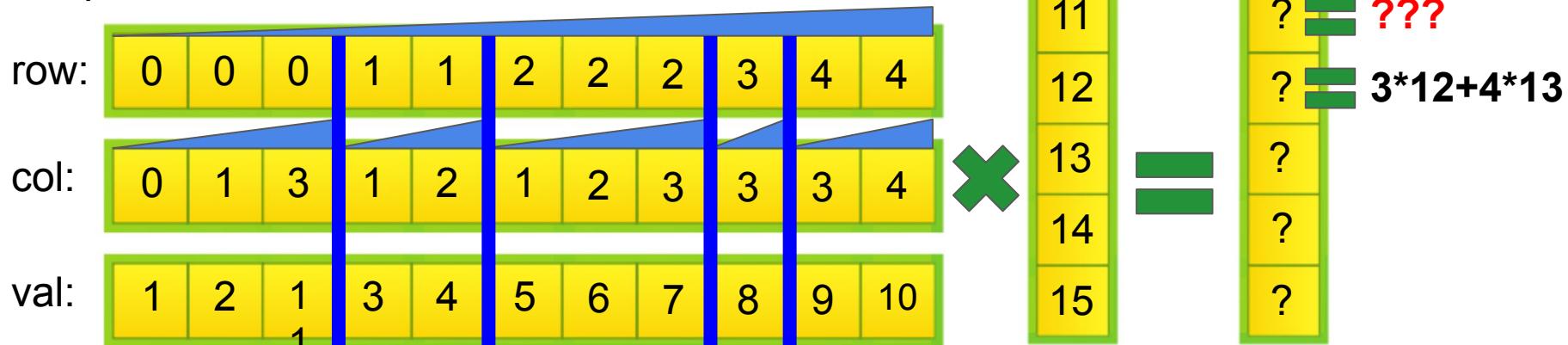


Насколько все хорошо?

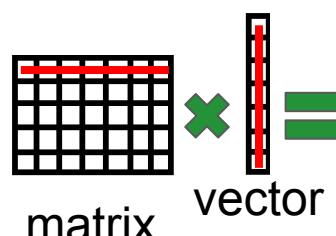
- компактность ($NNZ + NNZ + NNZ$)
- быстрое создание
- быстрое умножение?**

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

1) COOrdinate-wise - COO



SpMV (Sparse matrix–vector multiplication)

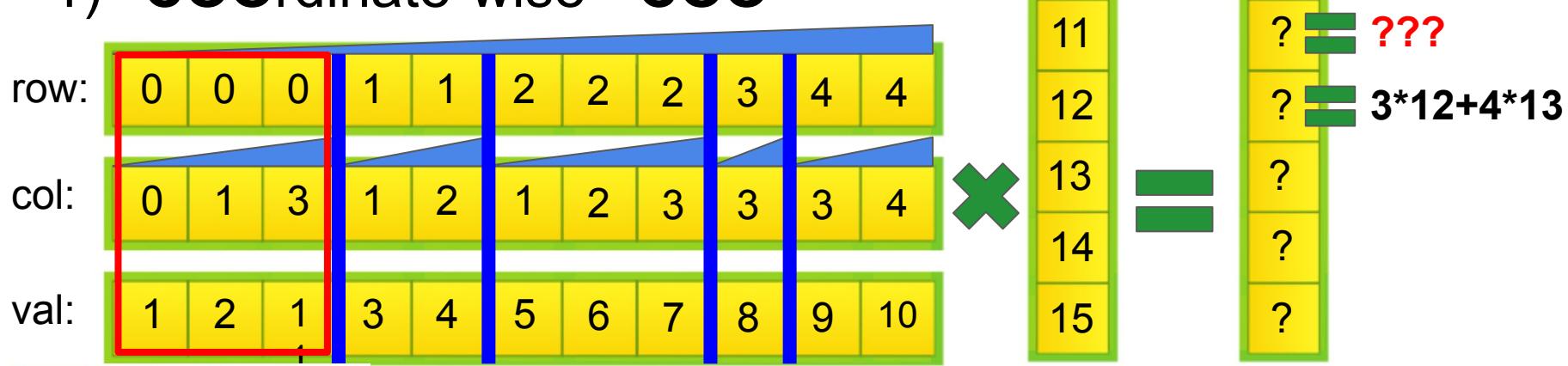


• **Насколько все хорошо?**

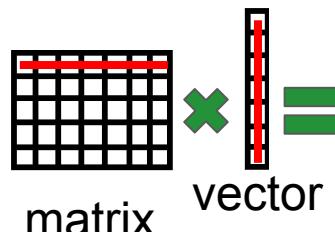
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- быстрое умножение?**

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

1) COORDINATE-WISE - COO



SpMV (Sparse matrix–vector multiplication)

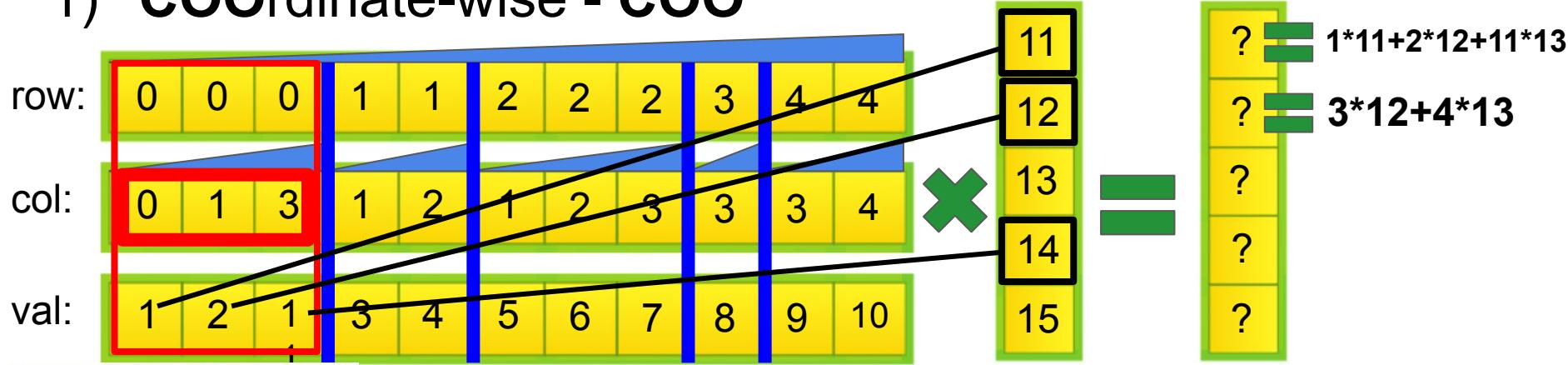


• Насколько все хорошо?

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

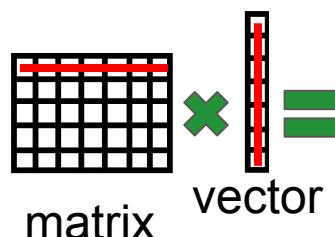
Источник: Sparse Matrix-Vector Multiplication and Matrix Formats

1) COOrdinate-wise - COO



1	2		11	
3	4			
5	6	7		
		8		
			9	10

SpMV (Sparse matrix–vector multiplication)

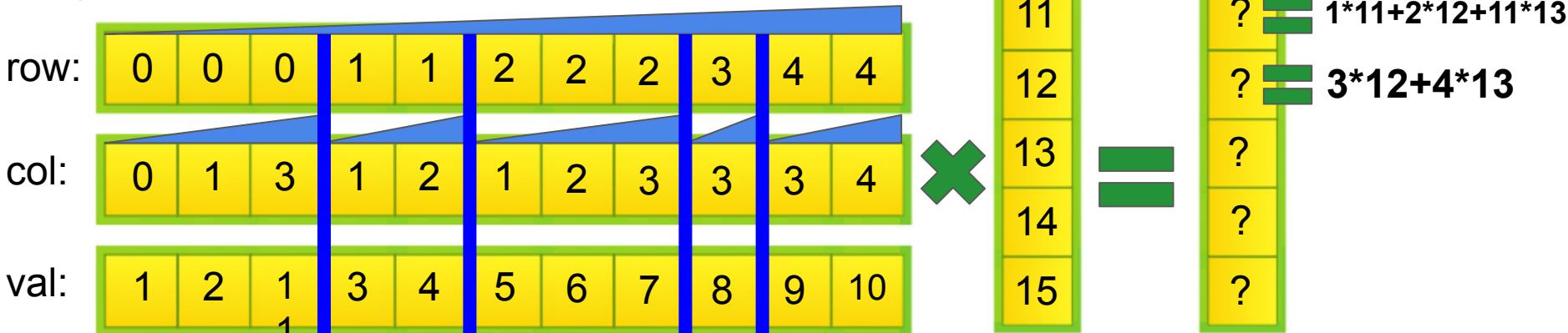


• **Насколько все хорошо?**

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

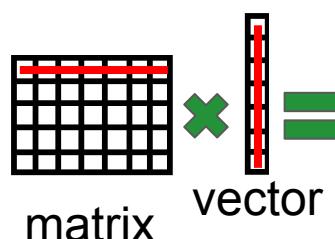
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

1) COOrdinate-wise - COO



1	2		11
3	4		
5	6	7	
			8
		9	10

SpMV (Sparse matrix–vector multiplication)

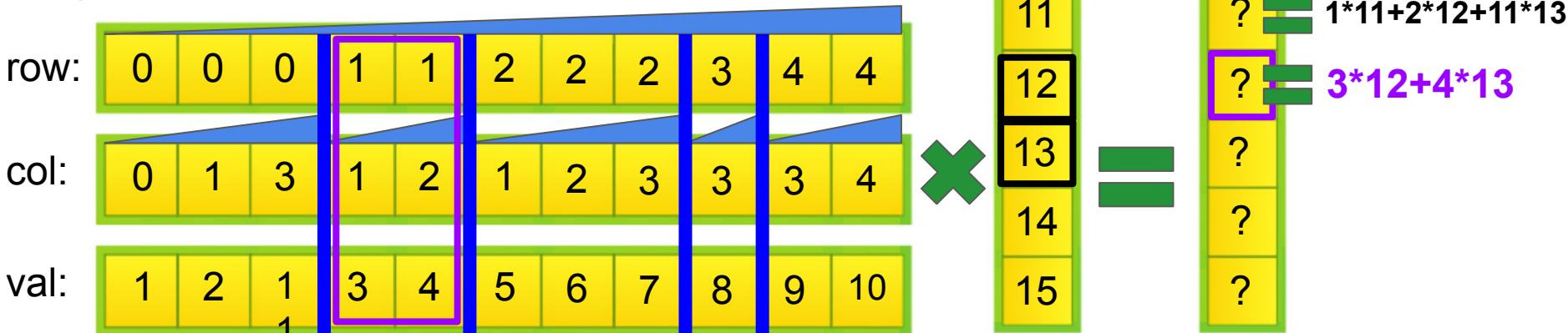


• Насколько все хорошо?

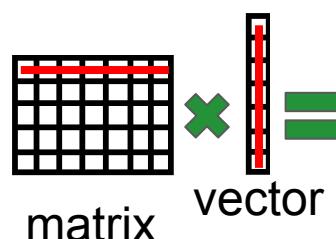
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- быстрое умножение?**

Что делает один WorkItem?

1) COOrdinate-wise - COO



SpMV (Sparse matrix–vector multiplication)



• **Насколько все хорошо?**

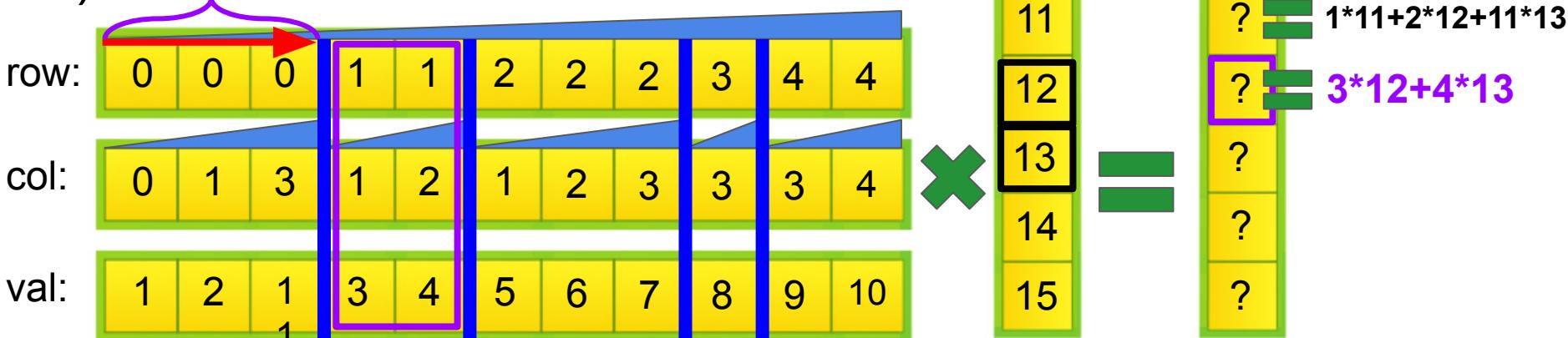
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- быстрое умножение?**

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

Как найти offset?

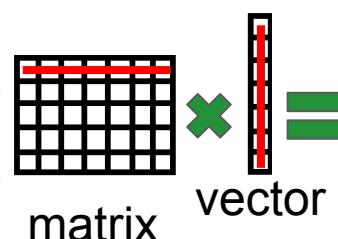
Что делает один WorkItem?

1) COOrdinate-wise - COO



1	2		11
3	4		
5	6	7	
			8
			9 10

SpMV (Sparse matrix–vector multiplication)



• Насколько все хорошо?

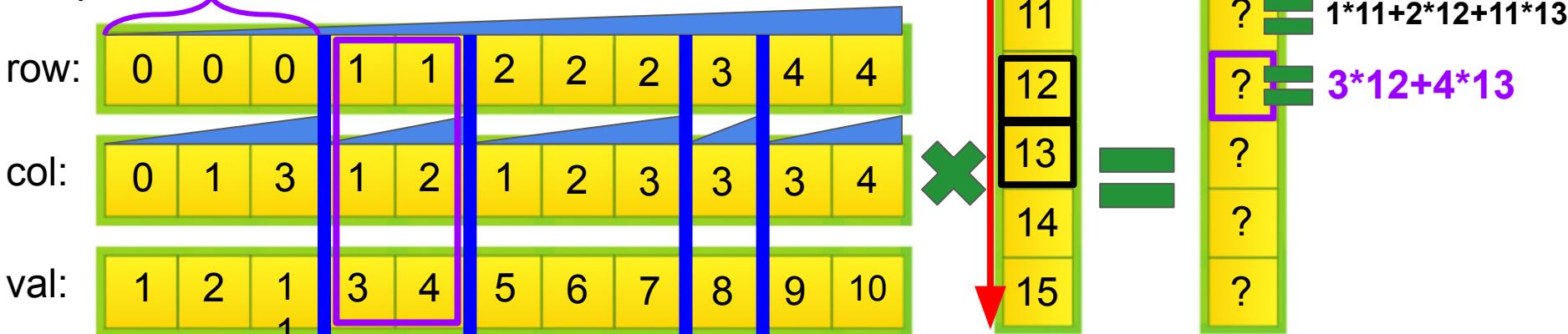
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- быстрое умножение?**

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

Как найти нужные элементы вектора?

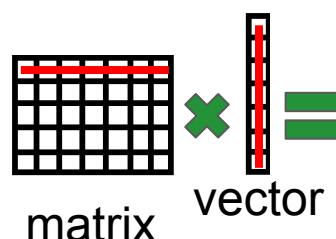
Что делает один WorkItem?

1) COOrdinate-wise - COO



1	2		11
3	4		
5	6	7	
			8
			9 10

SpMV (Sparse matrix–vector multiplication)



• Насколько все хорошо?

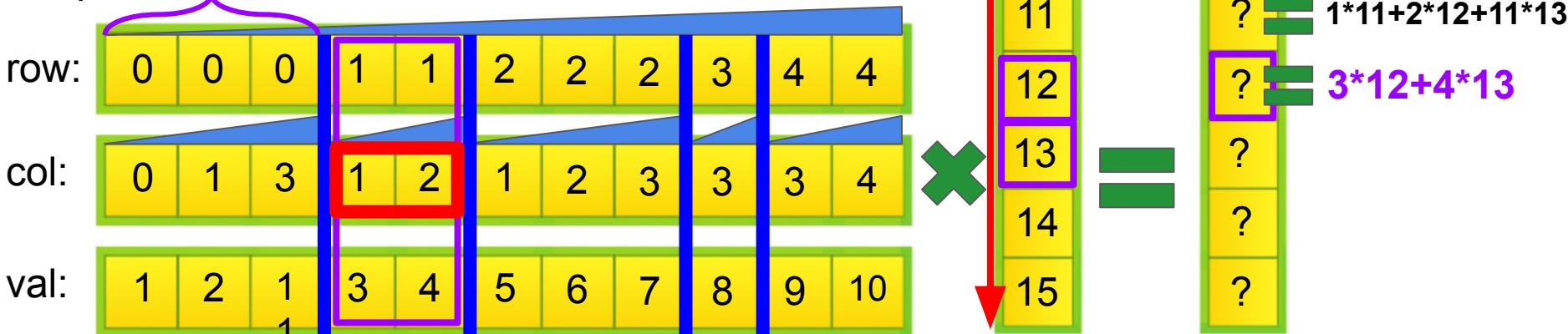
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- быстрое умножение?**

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

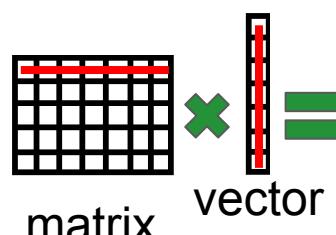
Как найти нужные элементы вектора?

Что делает один WorkItem?

1) COOrdinate-wise - COO



SpMV (Sparse matrix–vector multiplication)



• Насколько все хорошо?

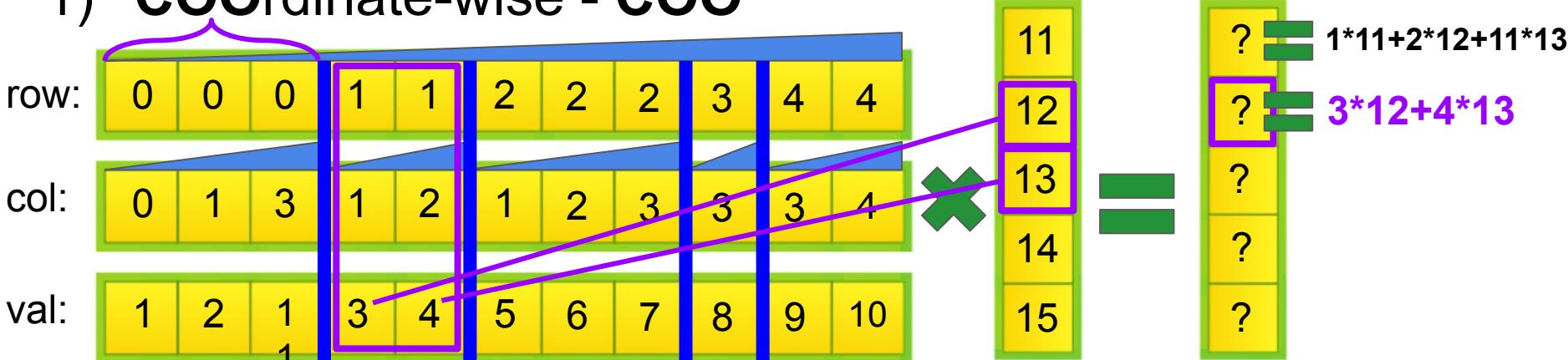
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- быстрое умножение?**

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

Как найти нужные элементы вектора?

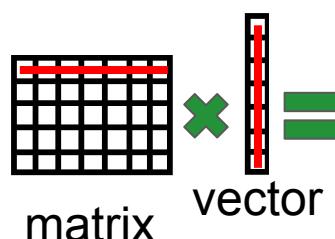
Что делает один WorkItem?

1) COOrdinate-wise - COO



1	2		11
3	4		
5	6	7	
		8	
		9	10

SpMV (Sparse matrix–vector multiplication)



• Насколько все хорошо?

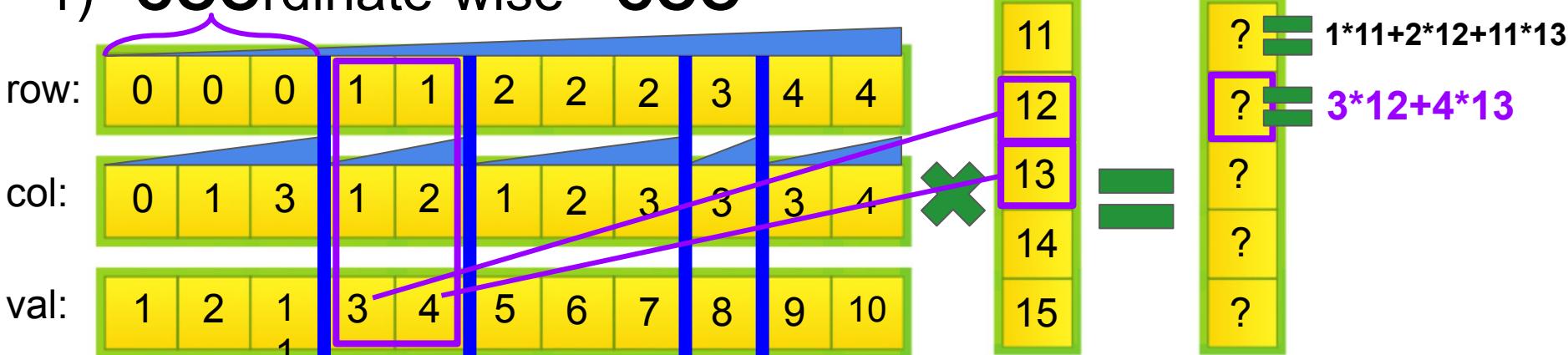
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- быстрое умножение?**

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

Что является узким местом?

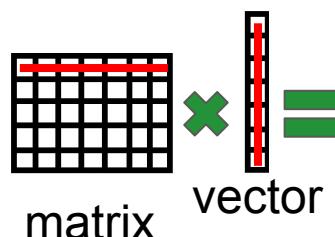
Что делает один WorkItem?

1) COOrdinate-wise - COO



SpMV (Sparse matrix–vector multiplication)

1	2	11								
3	4									
5	6	7								
		8								
		9	10							



• Насколько все хорошо?

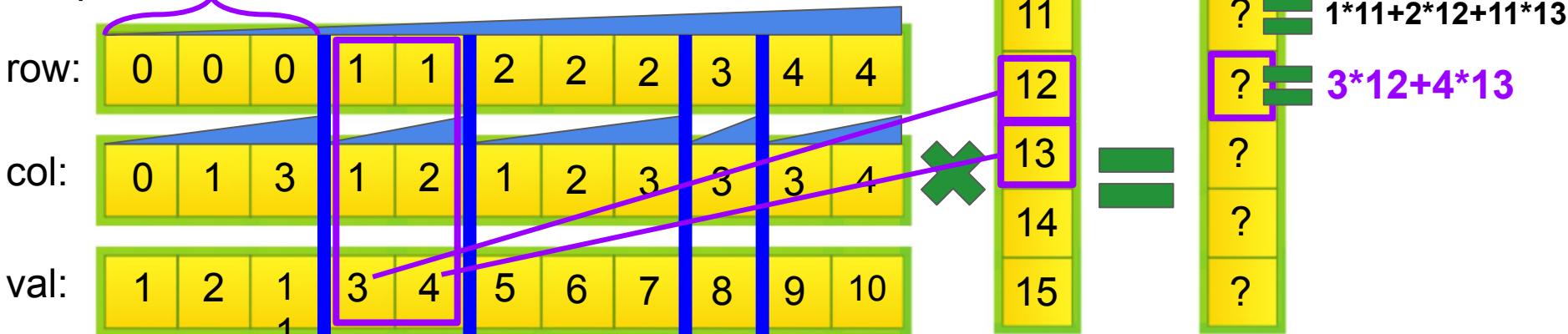
- компактность (**NNZ + NNZ + NNZ**)
- быстрое создание
- быстрое умножение?**

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

Что является узким местом?
Coalesced ли паттерн чтения?

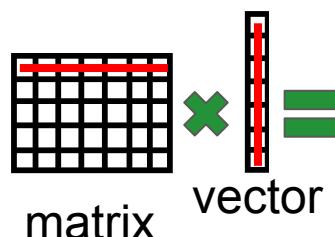
Что делает один WorkItem?

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:

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

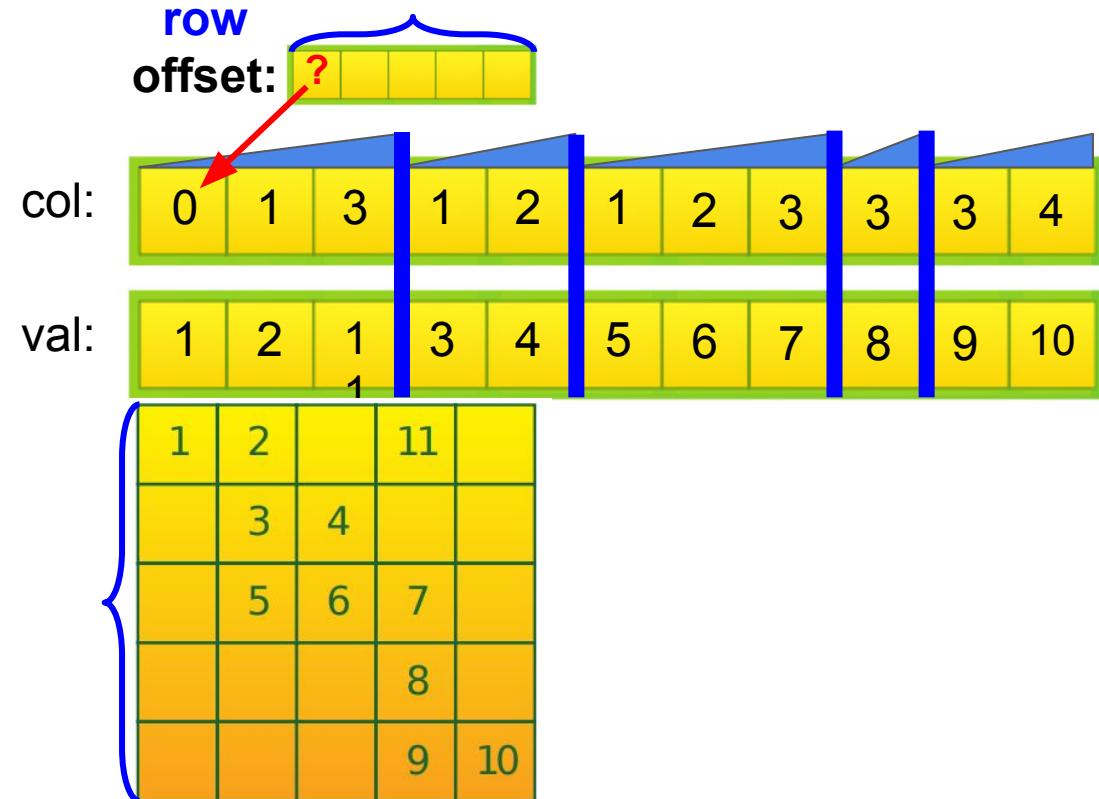
val:

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

1	2		11	
	3	4		
	5	6	7	
			8	
			9	10

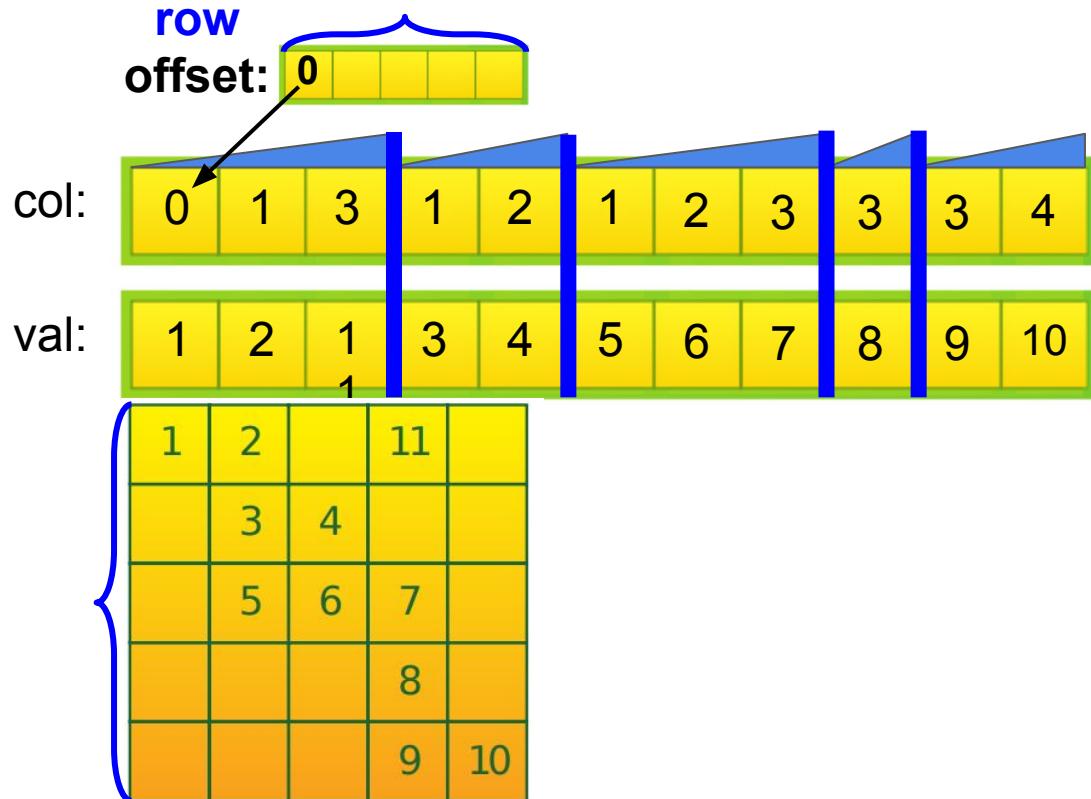
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

2) CSR - Compressed Sparse Row



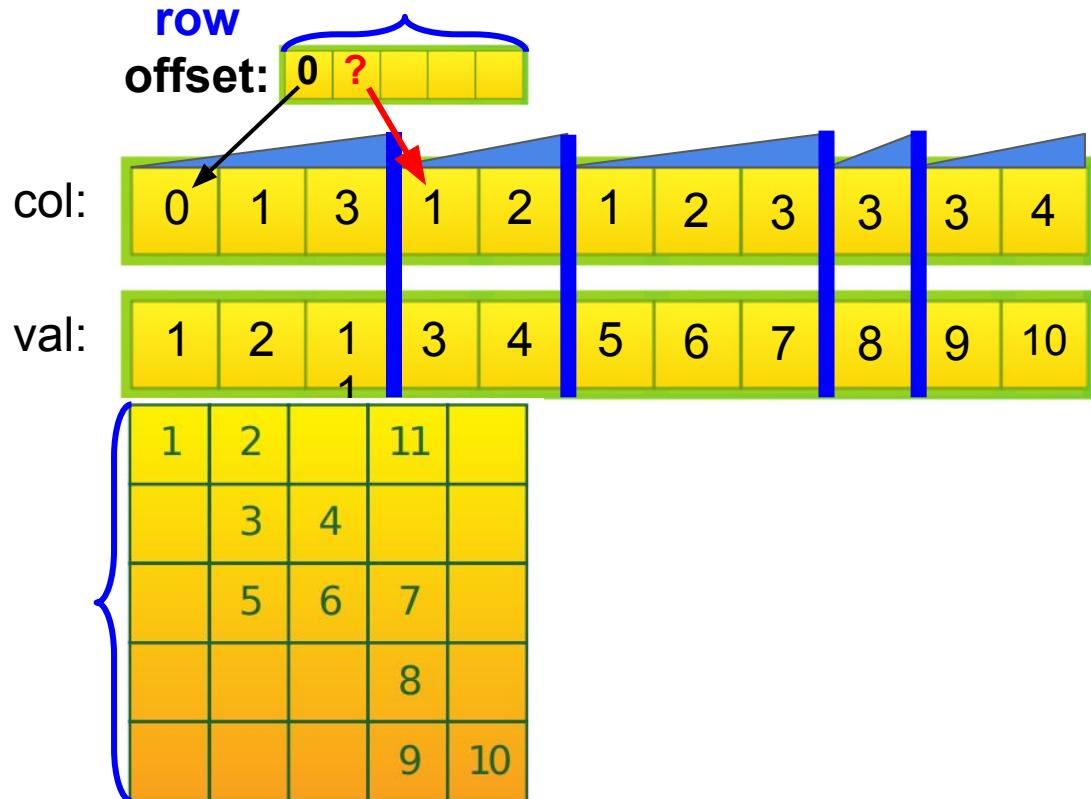
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

2) CSR - Compressed Sparse Row



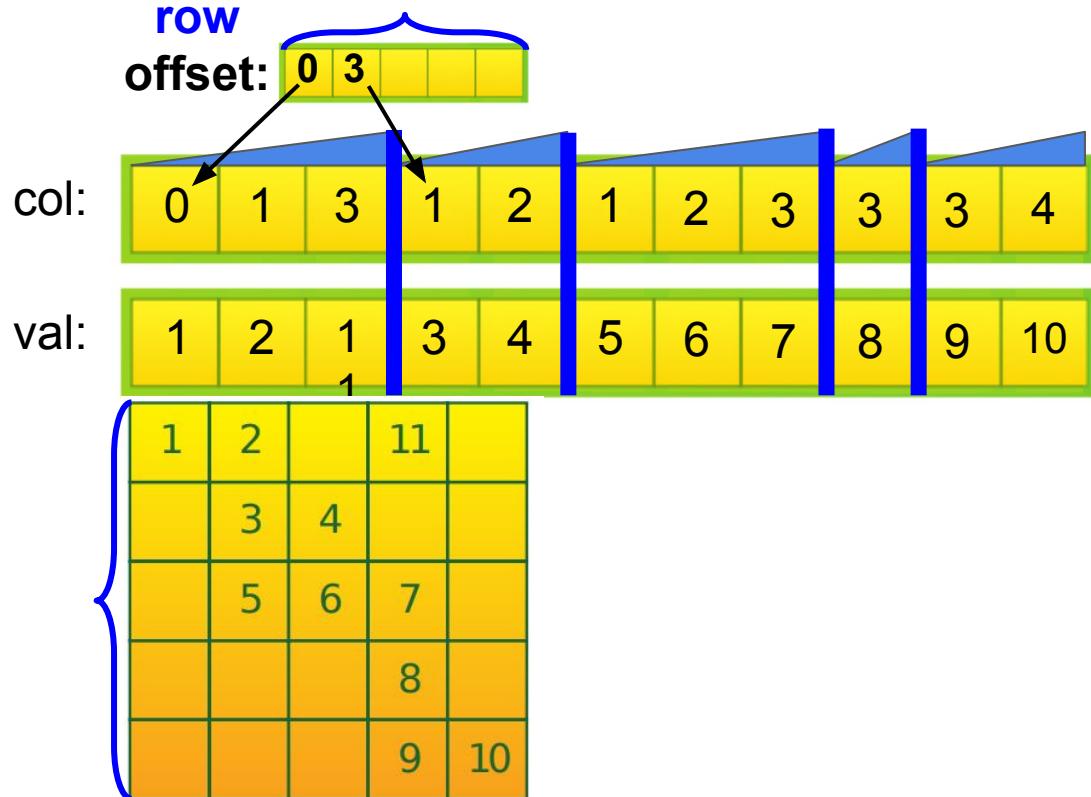
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

2) CSR - Compressed Sparse Row



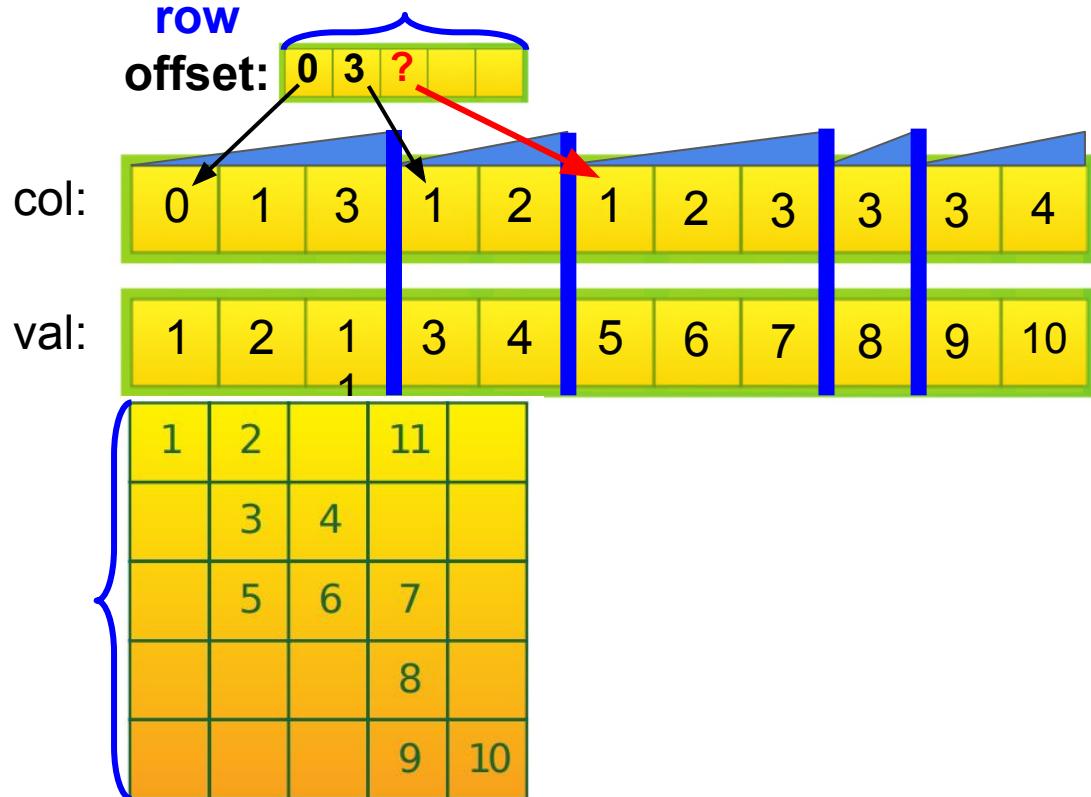
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

2) CSR - Compressed Sparse Row



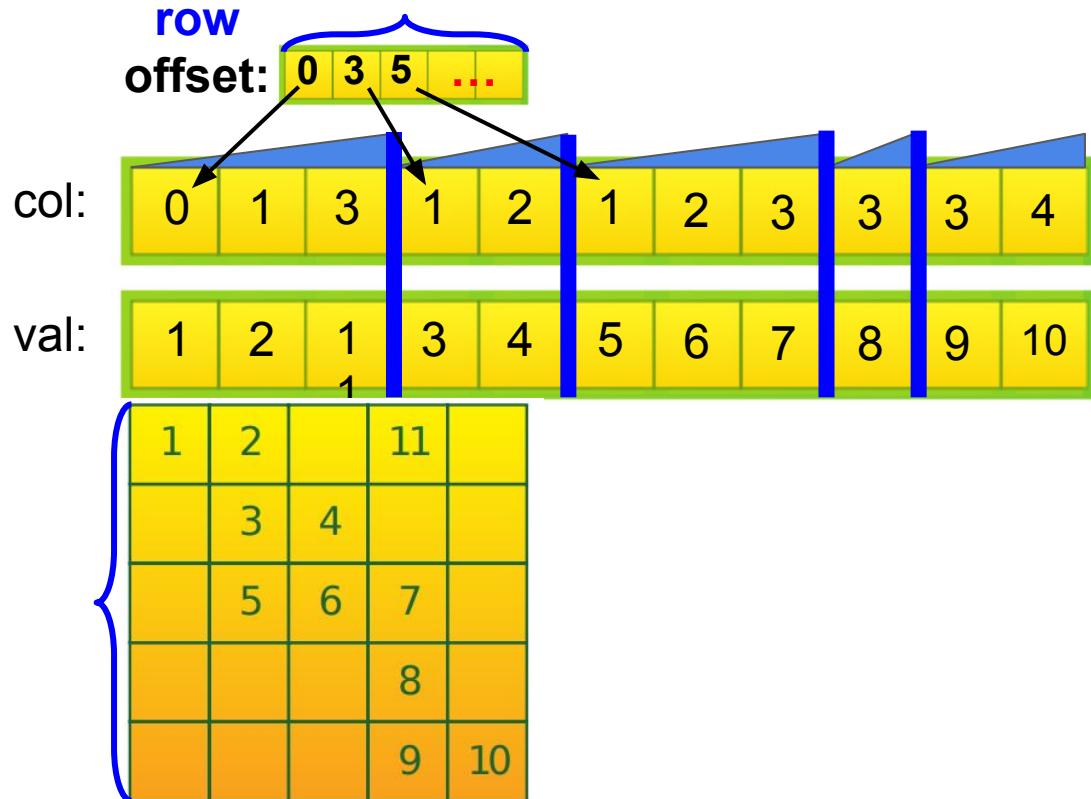
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

2) CSR - Compressed Sparse Row



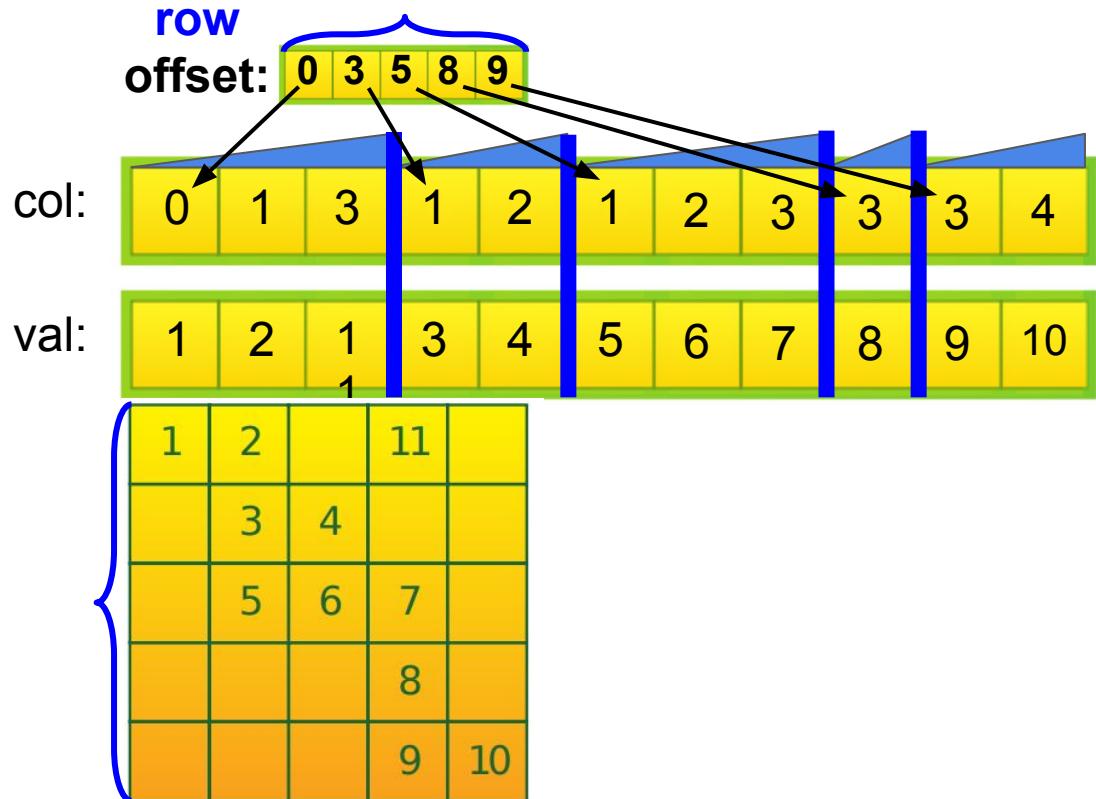
Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

2) CSR - Compressed Sparse Row



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

2) CSR - Compressed Sparse Row



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

2) CSR - Compressed Sparse Row



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

2) CSR - Compressed Sparse Row



Насколько все хорошо? - компактность?

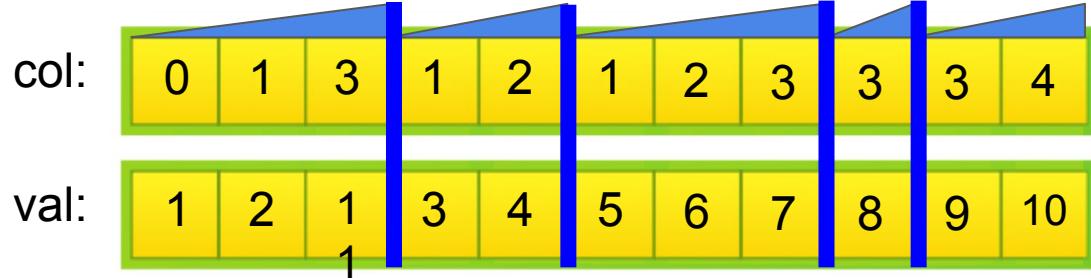
2) CSR - Compressed Sparse Row



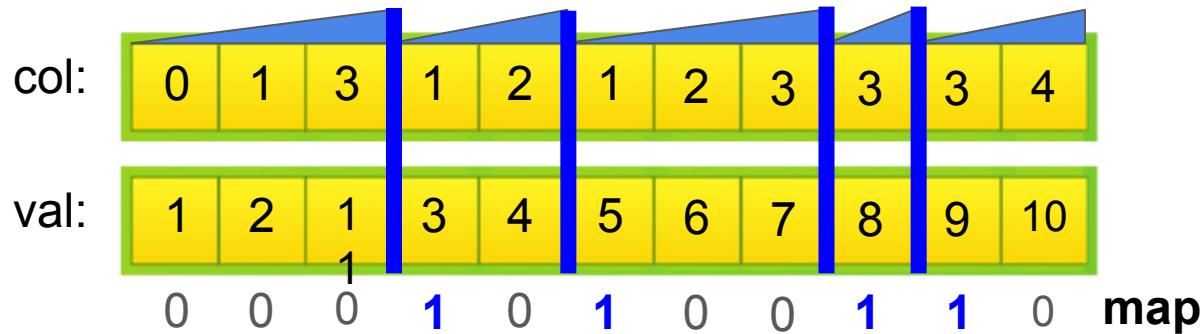
Насколько все хорошо?

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

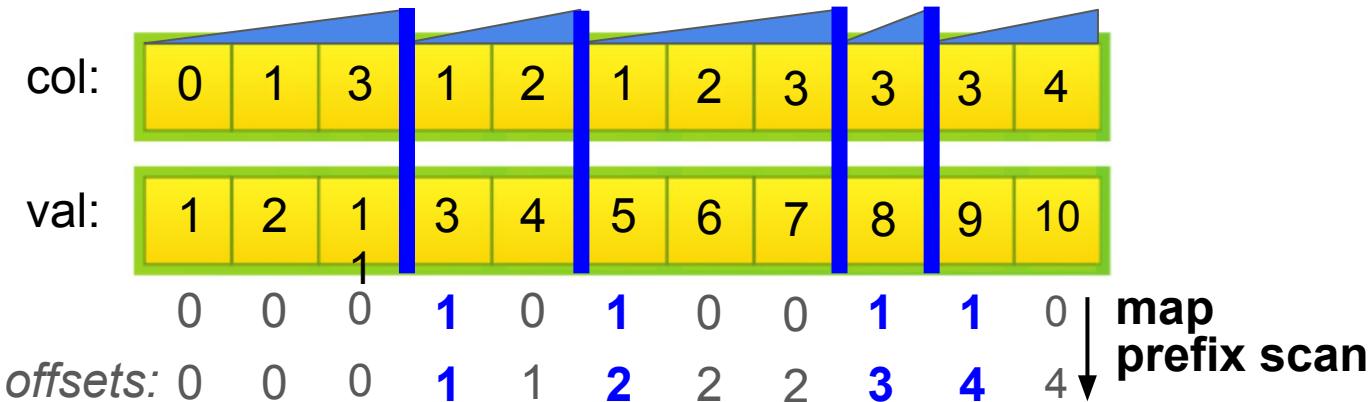
2) CSR - Compressed Sparse Row: **быстрое создание?**



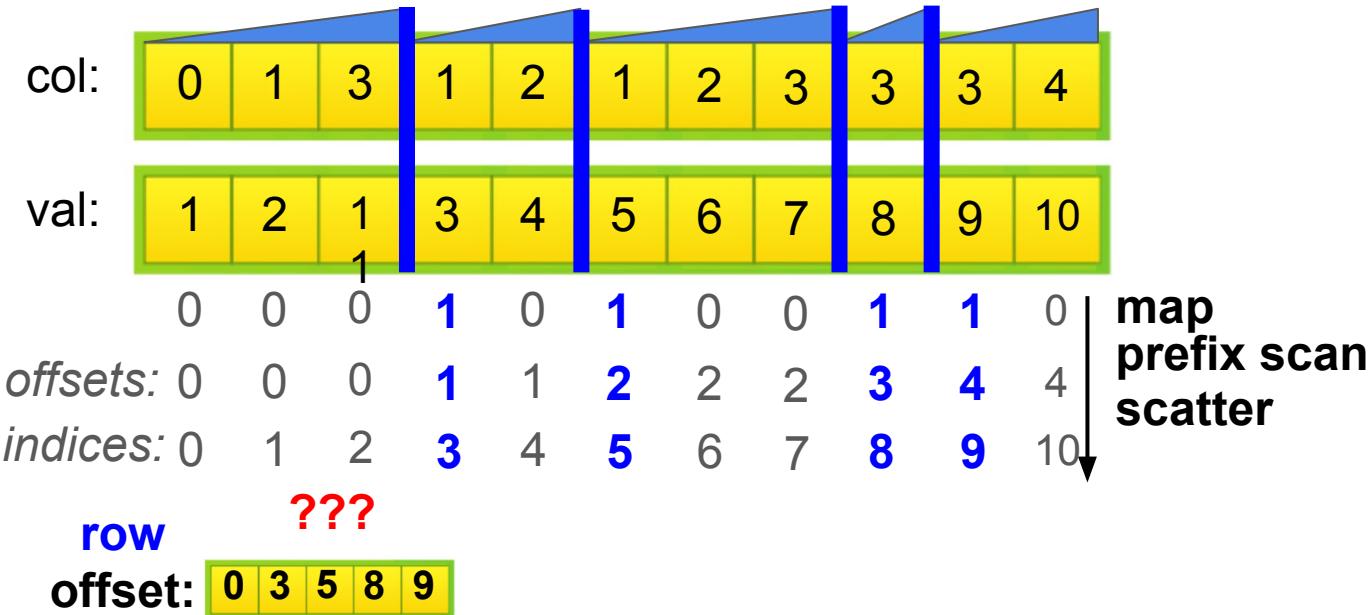
2) CSR - Compressed Sparse Row: быстрое создание?



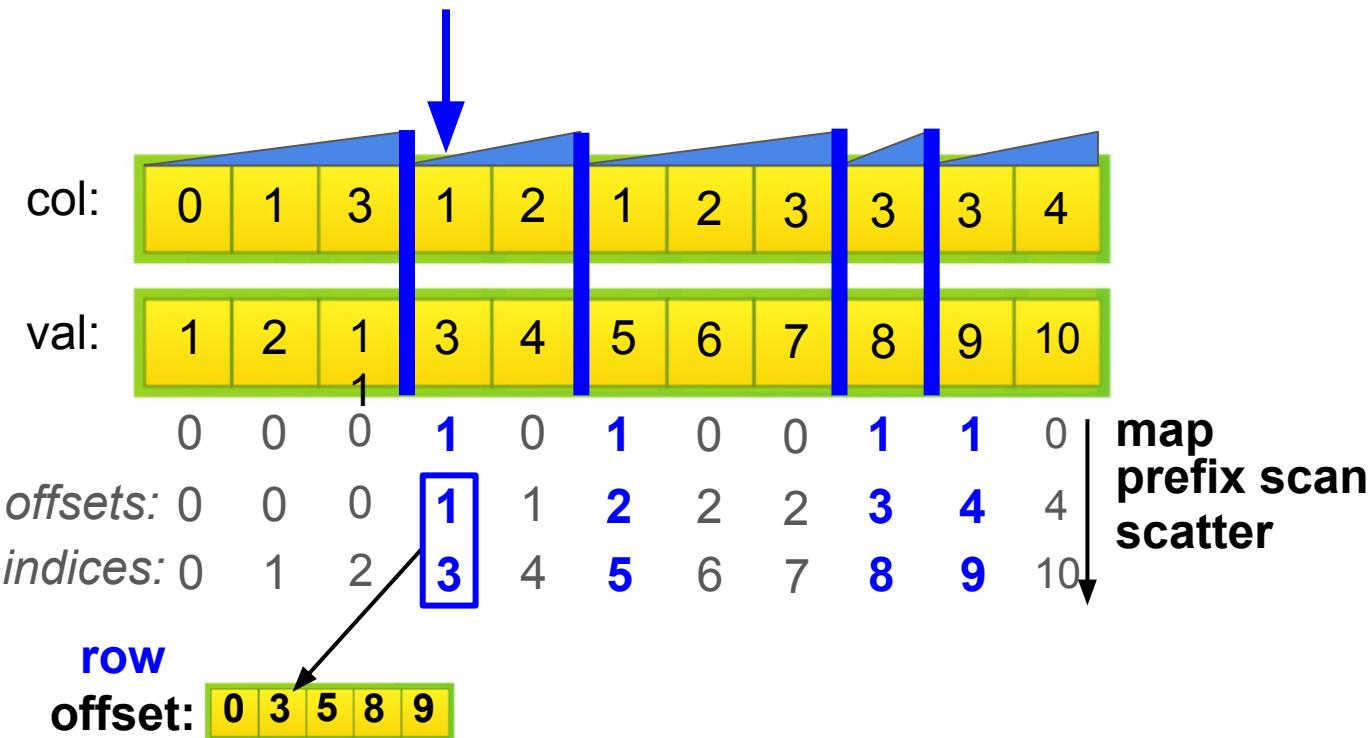
2) CSR - Compressed Sparse Row: быстрое создание?



2) CSR - Compressed Sparse Row: быстрое создание?

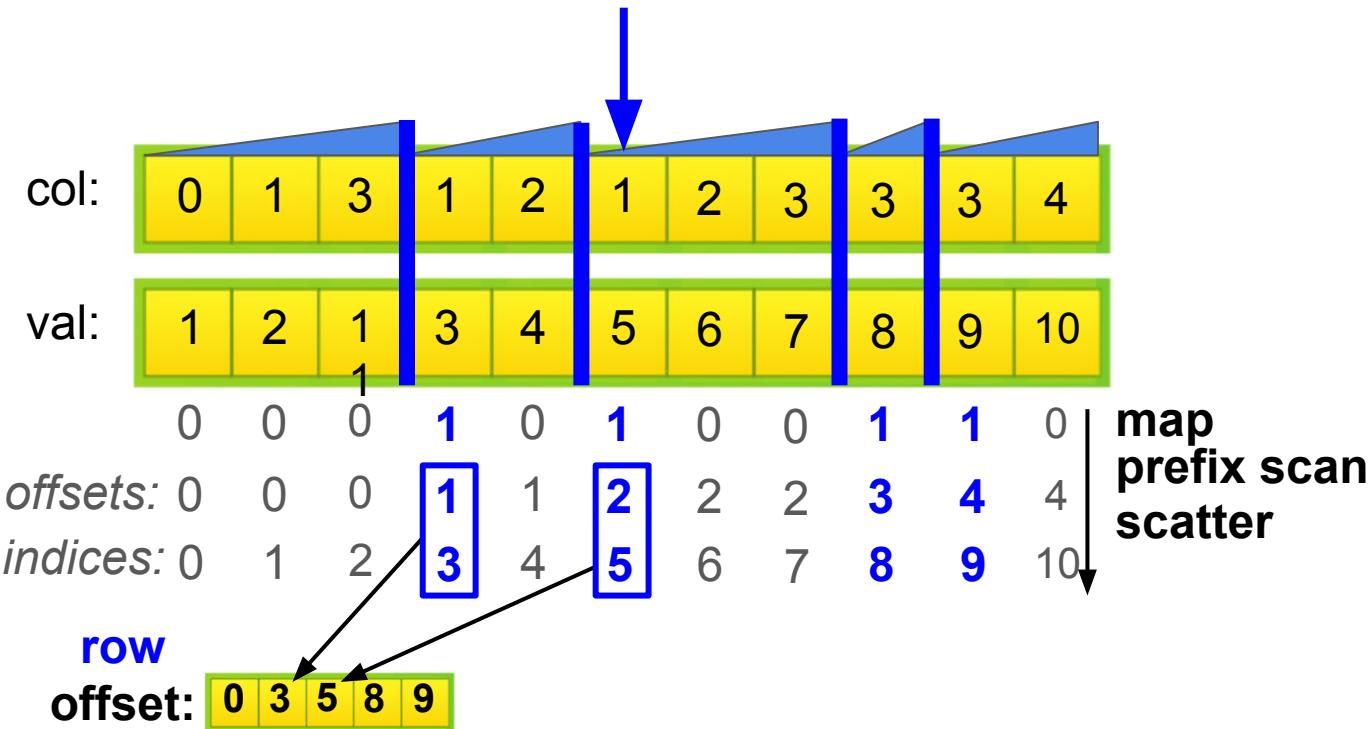


2) CSR - Compressed Sparse Row: быстрое создание?

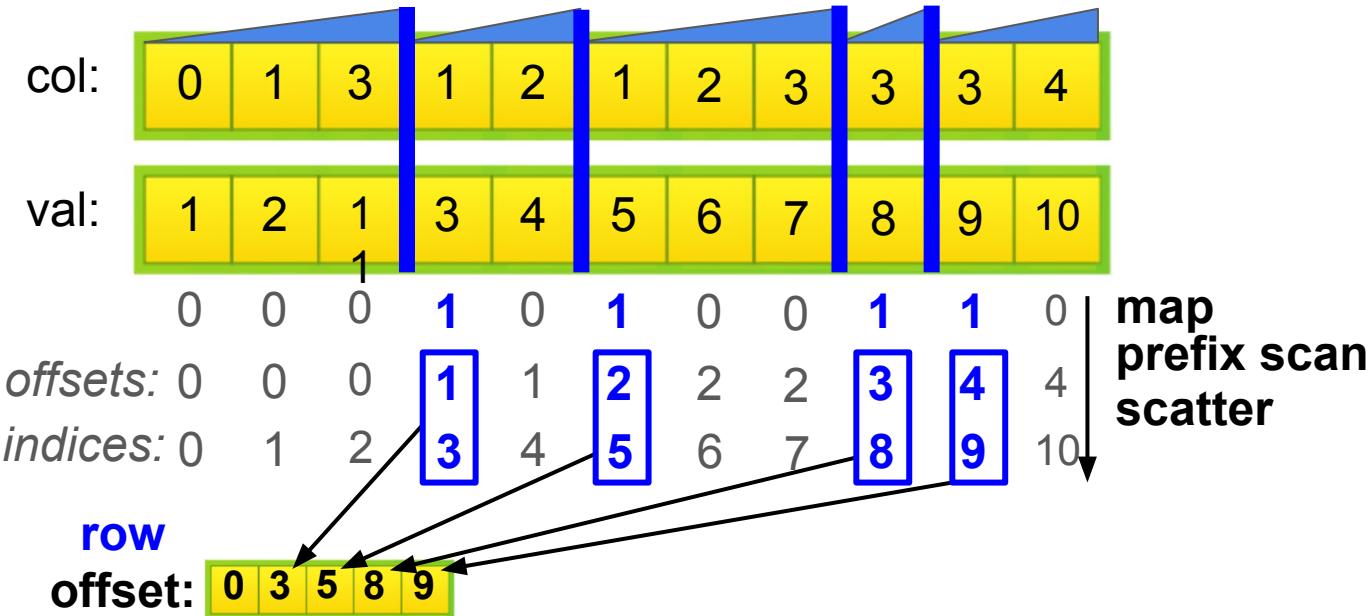


Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)

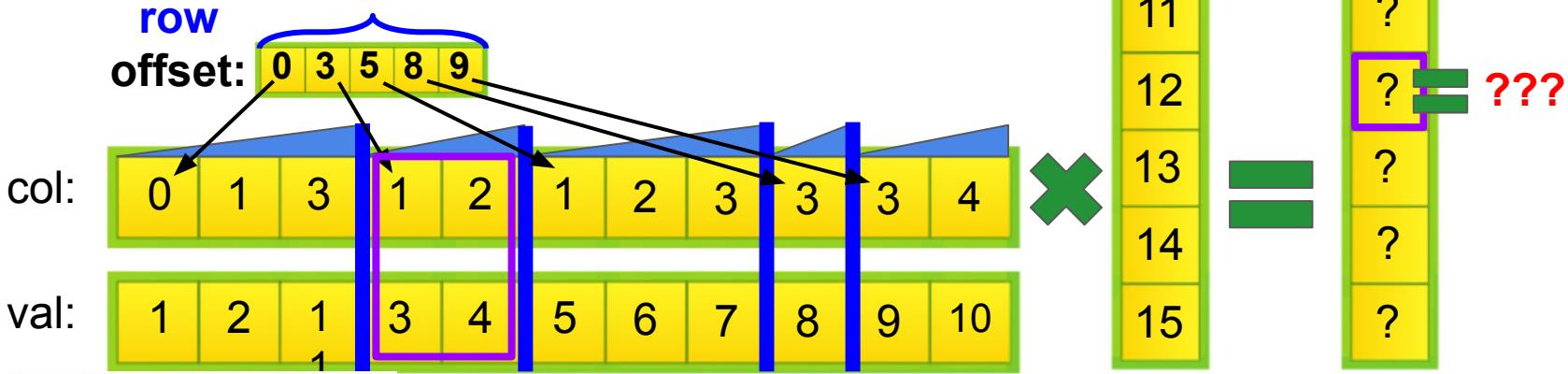
2) CSR - Compressed Sparse Row: быстрое создание?



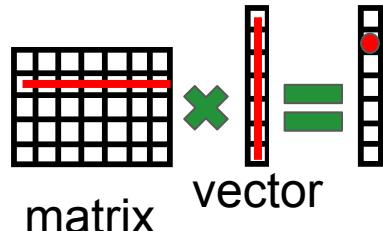
2) CSR - Compressed Sparse Row: быстрое создание?



2) CSR - Compressed Sparse Row



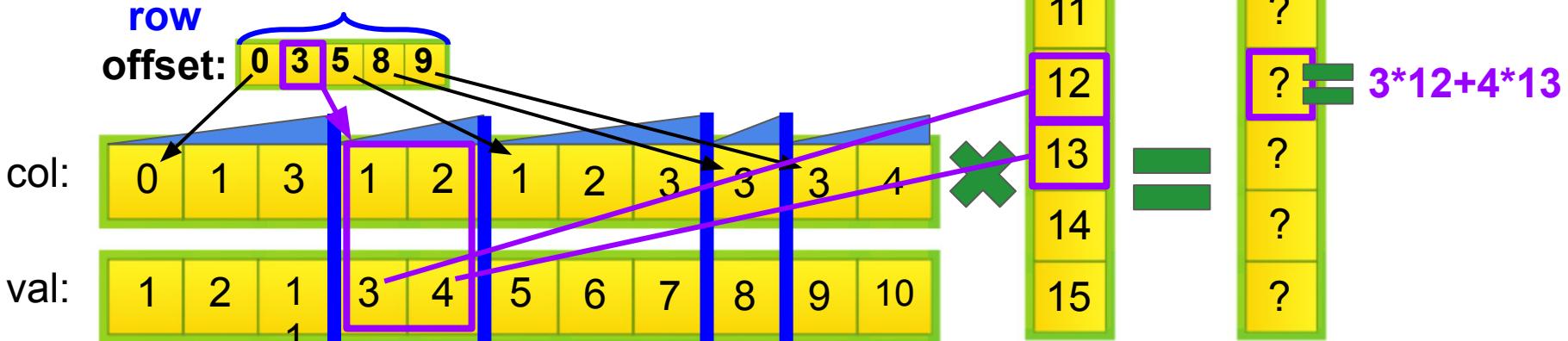
1	2		11	
3	4			
5	6	7		
			8	
			9	10



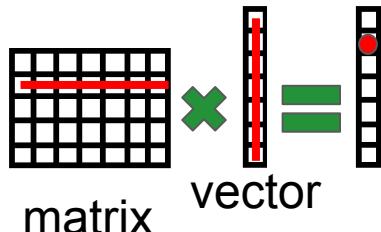
Насколько все хорошо?

- компактность ($N + \text{NNZ} + \text{NNZ}$)
- быстрое создание
- быстрое умножение?
что делает WorkGroup?

2) CSR - Compressed Sparse Row



1	2		11	
3	4			
5	6	7		
			8	
			9	10

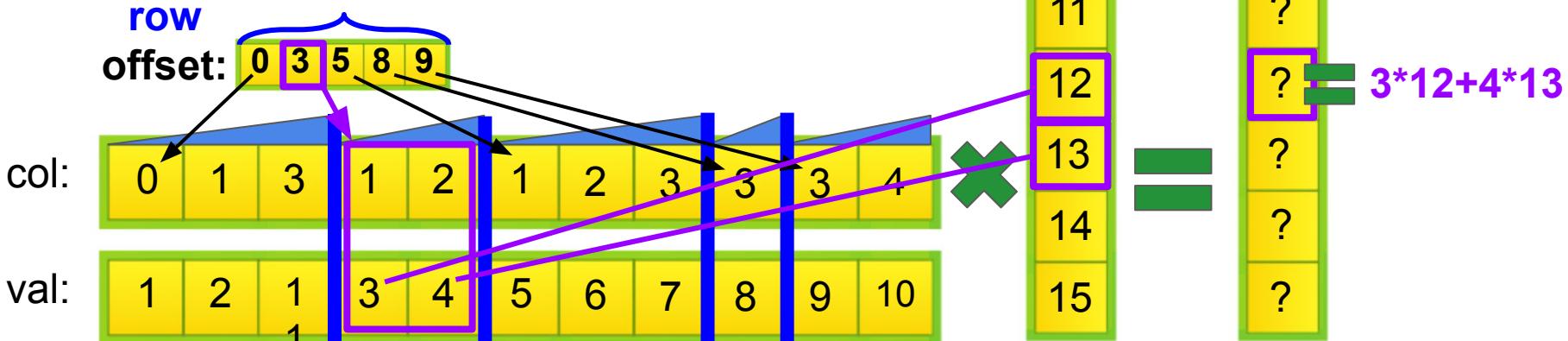


Насколько все хорошо?

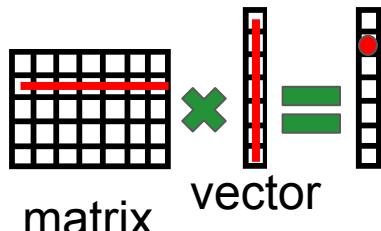
- компактность ($N + \text{NNZ} + \text{NNZ}$)
- быстрое создание
- быстрое умножение?
что делает WorkGroup?

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

2) CSR - Compressed Sparse Row



1	2		11	
3	4			
5	6	7		
			8	
			9	10

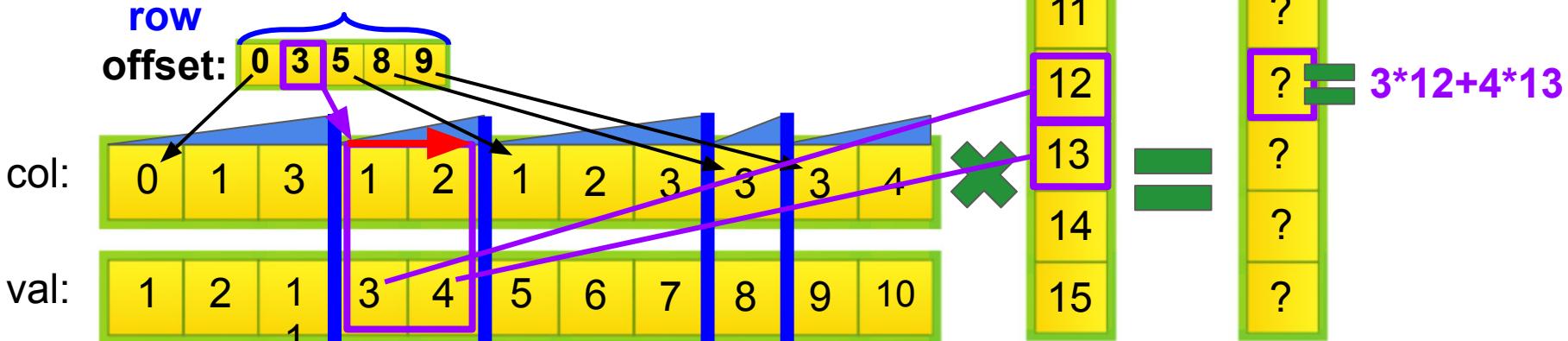


Насколько все хорошо?

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

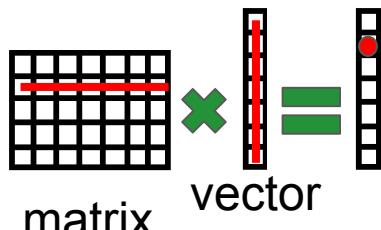
Почему именно по рядам сжимаем?

2) CSR - Compressed Sparse Row



Насколько все хорошо?

- компактность ($N + \text{NNZ} + \text{NNZ}$)
- быстрое создание
- быстрое умножение (лучше чем COO)



3) **CSC** - Compressed Sparse Column

Такие же Non-Zero Values как в CSR?

val:	1	2	1	3	4	5	6	7	8	9	10
	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	1	3	4	5	6	7	8	9	10
1	2		11								
	3	4									
	5	6	7								
			8								
				9	10						

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

3) **CSC** - 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

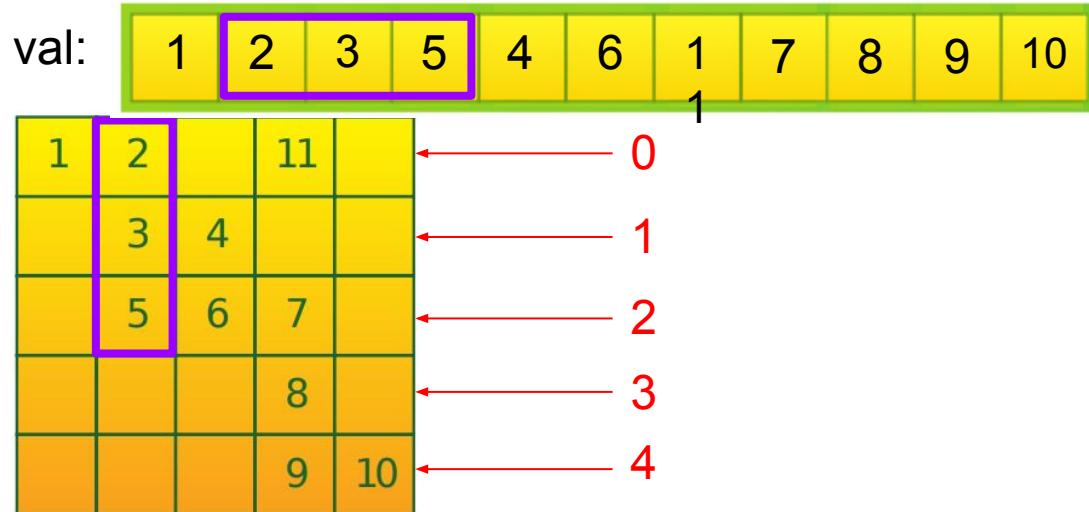
val: 

1	2		11	
	3	4		
	5	6	7	
			8	
			9	10

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

3) CSC- Compressed Sparse Column

Как выглядит row?



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

3) CSC- Compressed Sparse Column

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

0

1

2

3

4

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
------	---	---	---	---	---	---	---	---	---	---	----

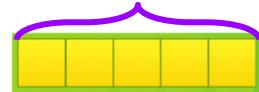
1	2		11	
	3	4		
	5	6	7	
			8	
			9	10

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

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	2		11	
	3	4		
	5	6	7	
			8	
			9	10

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

3) CSC - Compressed Sparse Column

column

offset:



row:



val:



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

3) CSC - Compressed Sparse Column

column

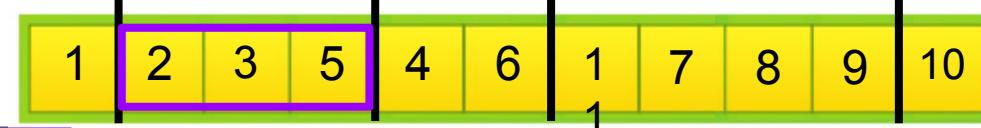
offset:



row:



val:



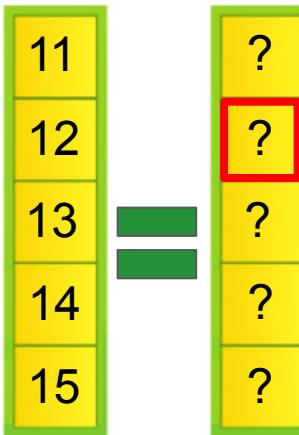
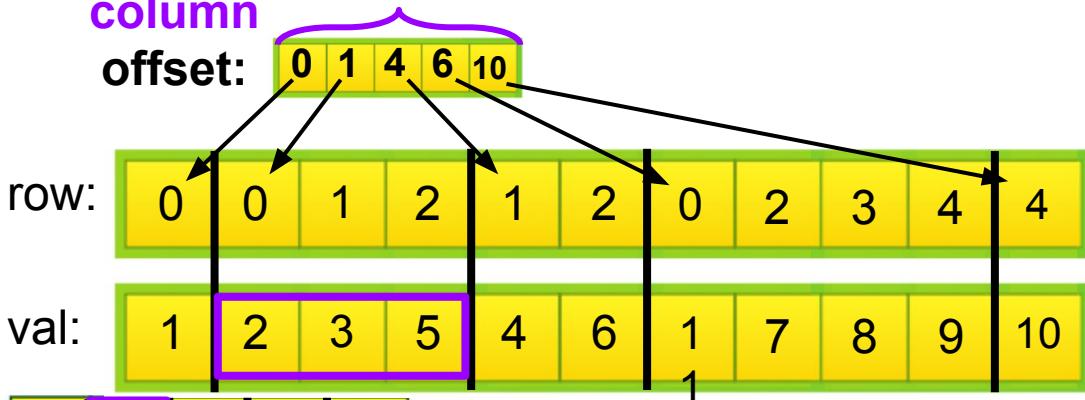
Насколько все хорошо?

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

3) CSC - Compressed Sparse Column

column

offset:



Что из матрицы
нужно?



Насколько все хорошо?

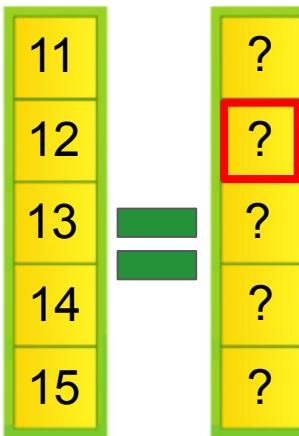
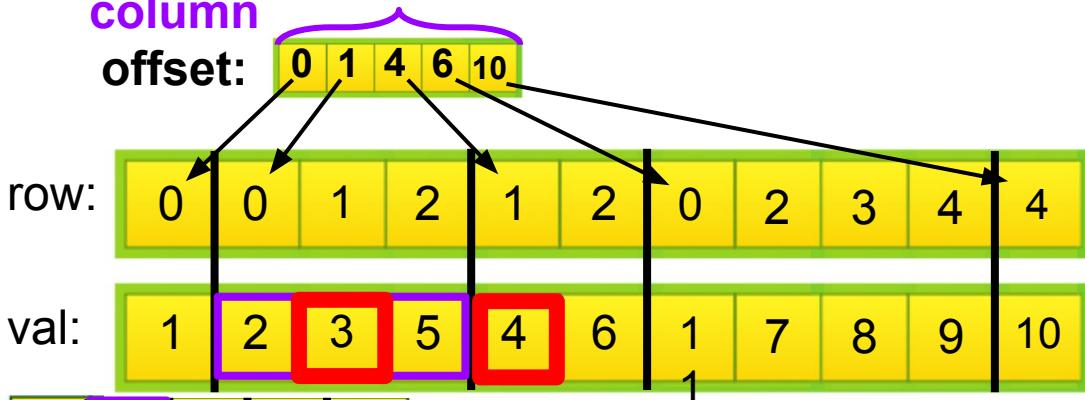
- компактность ($N + NNZ + NNZ$)
- быстрое создание
- быстрое умножение?

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

3) CSC - Compressed Sparse Column

column

offset:



Насколько все хорошо?

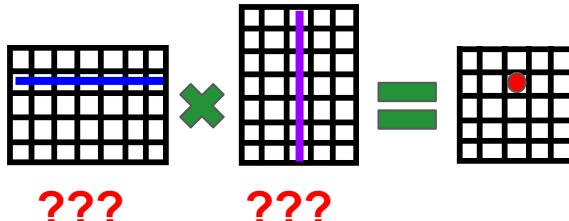
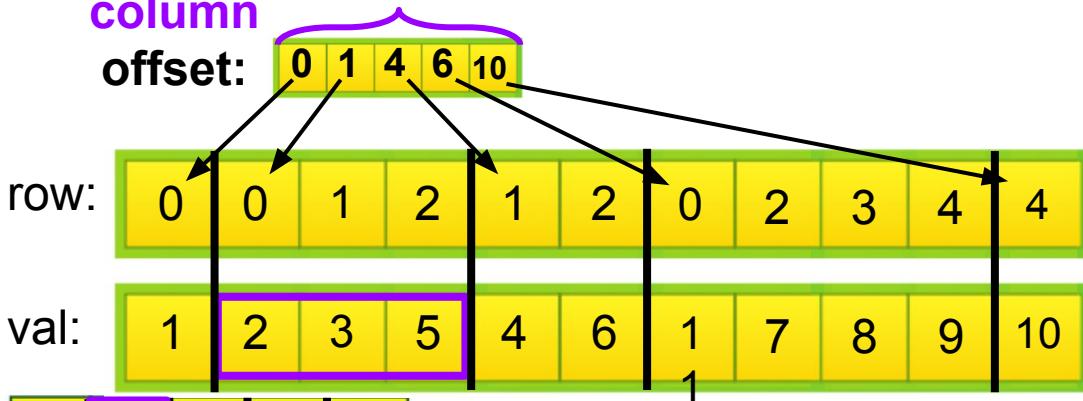
- компактность ($N + NNZ + NNZ$)
- быстрое создание
- быстрое умножение?

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

3) CSC - Compressed Sparse Column

column

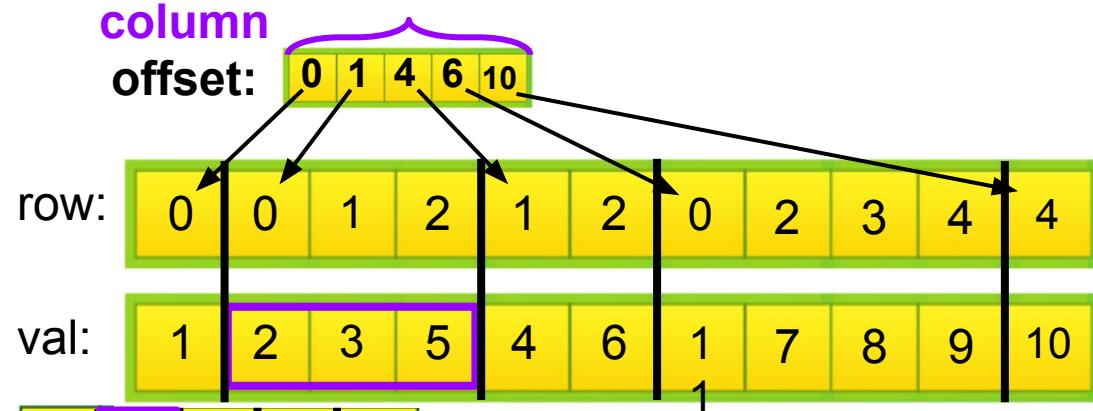
offset:



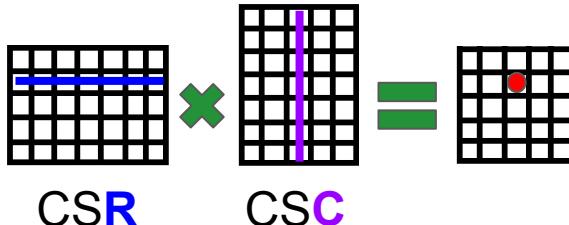
Насколько все хорошо?

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

3) CSC - Compressed Sparse Column



1	2	11	
3	4		
5	6	7	
		8	
		9	10



Насколько все хорошо?

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

Использование

ELLpack, DIAGONAL, HYBRID

Глава 2

4) ELL - ELLPACK

1	2		11	
	3	4		
5	6	7		
			8	
			9	10

Сколько максимум NNZ элементов в ряде?

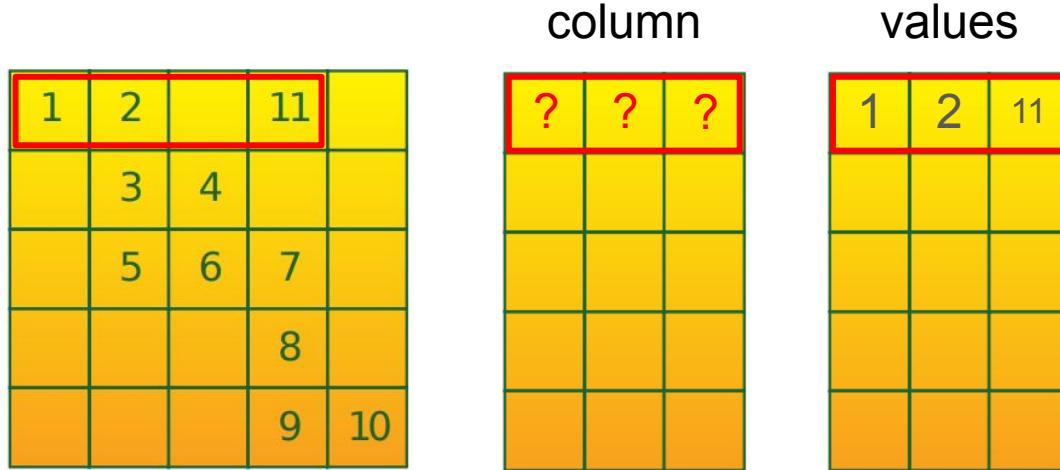
4) ELL - ELLPACK

1	2		11	
	3	4		
5	6	7		
		8		
		9	10	

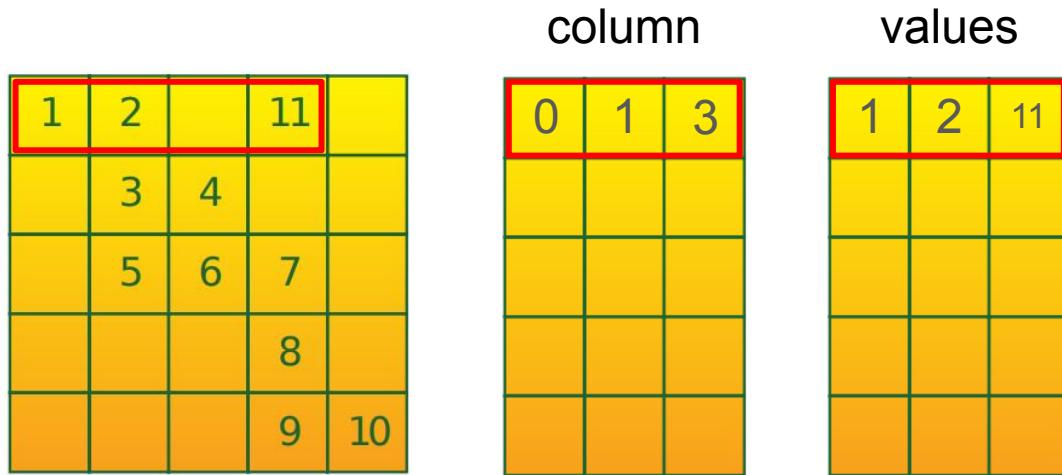
values

1	2	11

4) ELL - ELLPACK



4) ELL - ELLPACK



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?

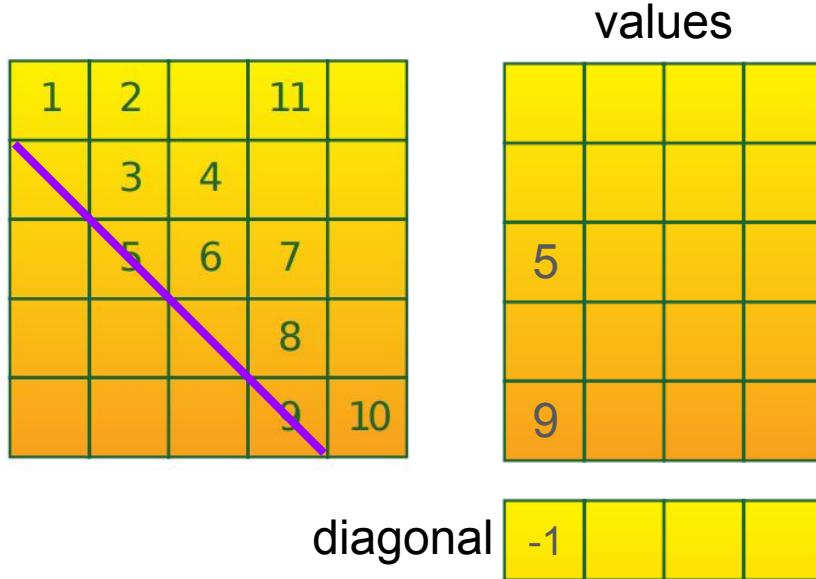
- компактность ($N \cdot K + N \cdot K$), хоть и хуже чем $N + NNZ + NNZ$
- зато **равномерность рабочей нагрузки** если везде $\sim K$
- быстрое создание
- быстрое умножение

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

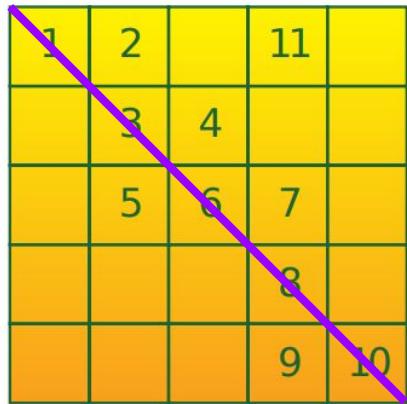
5) DIA - DIAGONAL **идеи?**

1	2		11	
	3	4		
	5	6	7	
			8	
			9	10

5) DIA - DIAGONAL



5) DIA - DIAGONAL



1	2		11	
	3	4		
	5	6	7	
		8		
		9	10	

values

	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	
		8	
9	10		

diagonal

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

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
----	---	----	----

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 ($D^*N + D$), хуже чем $2*NNZ$
- 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 ($D^*N + D$), хуже чем $2*NNZ$
- зато **равномерность рабочей нагрузки**
- быстрое умножение

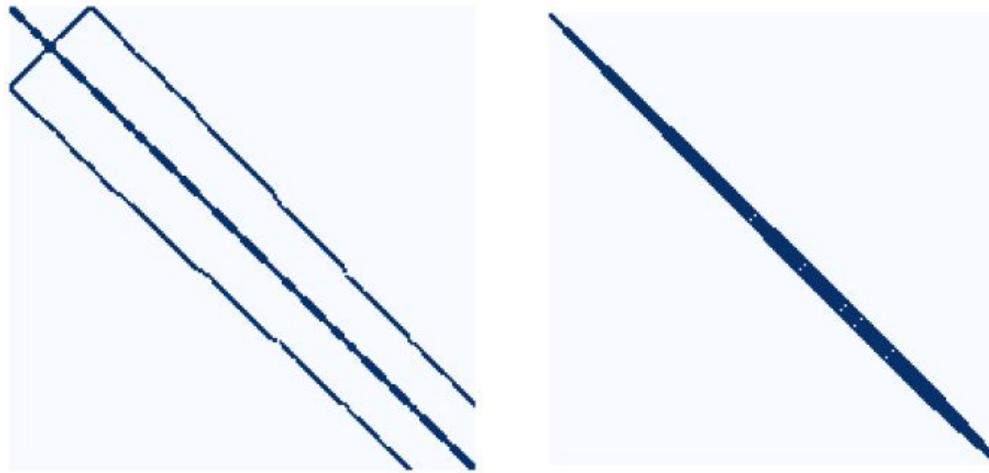
Format	Structure	Values
Dense	—	$N \times N$
COO	$2 \times NNZ$	NNZ
CSR	$N + 1 + NNZ$	NNZ
ELL	$M \times N$	$M \times N$
DIA	D	$D \times N_D$

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

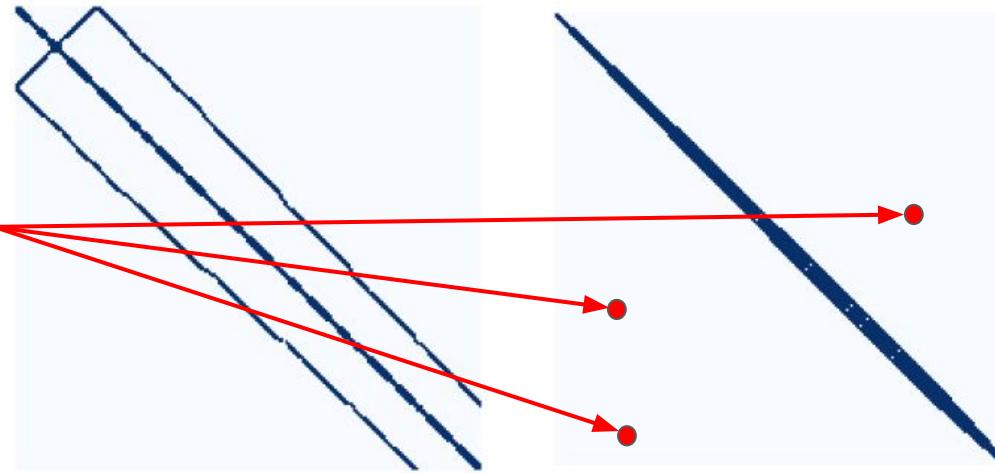


Format	Structure	Values
Dense	–	$N \times N$
COO	$2 \times NNZ$	NNZ
CSR	$N + 1 + NNZ$	NNZ
ELL	$M \times N$	$M \times N$
DIA	D	$D \times N_D$

Когда хороши?

5) DIA - DIAGONAL

А если небольшая доля выбросов?

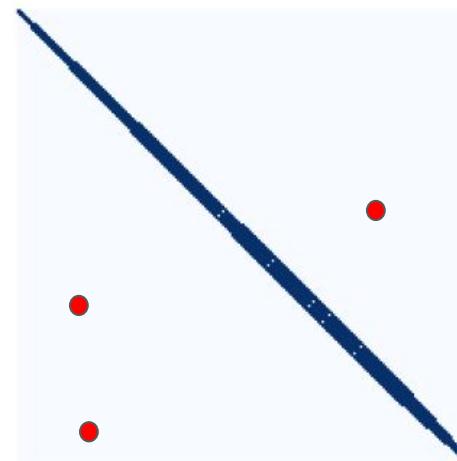


Format	Structure	Values
Dense	–	$N \times N$
COO	$2 \times NNZ$	NNZ
CSR	$N + 1 + NNZ$	NNZ
ELL	$M \times N$	$M \times N$
DIA	D	$D \times N_D$

Когда хороши?

6) HYB = ELL/DIA + COO

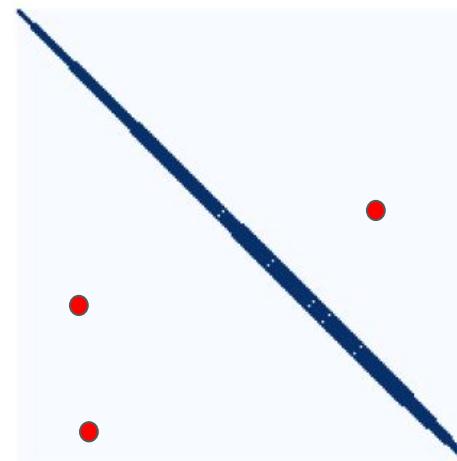
$A = B$ + C (для выбросов)



6) HYB = ELL/DIA + COO

$A = B$ + C (для выбросов)

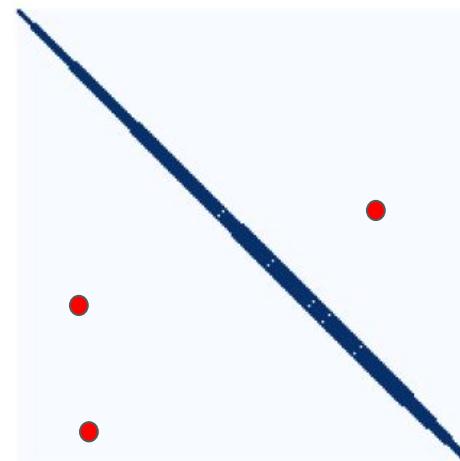
Как сделать Axv ?



6) HYB = ELL/DIA + COO

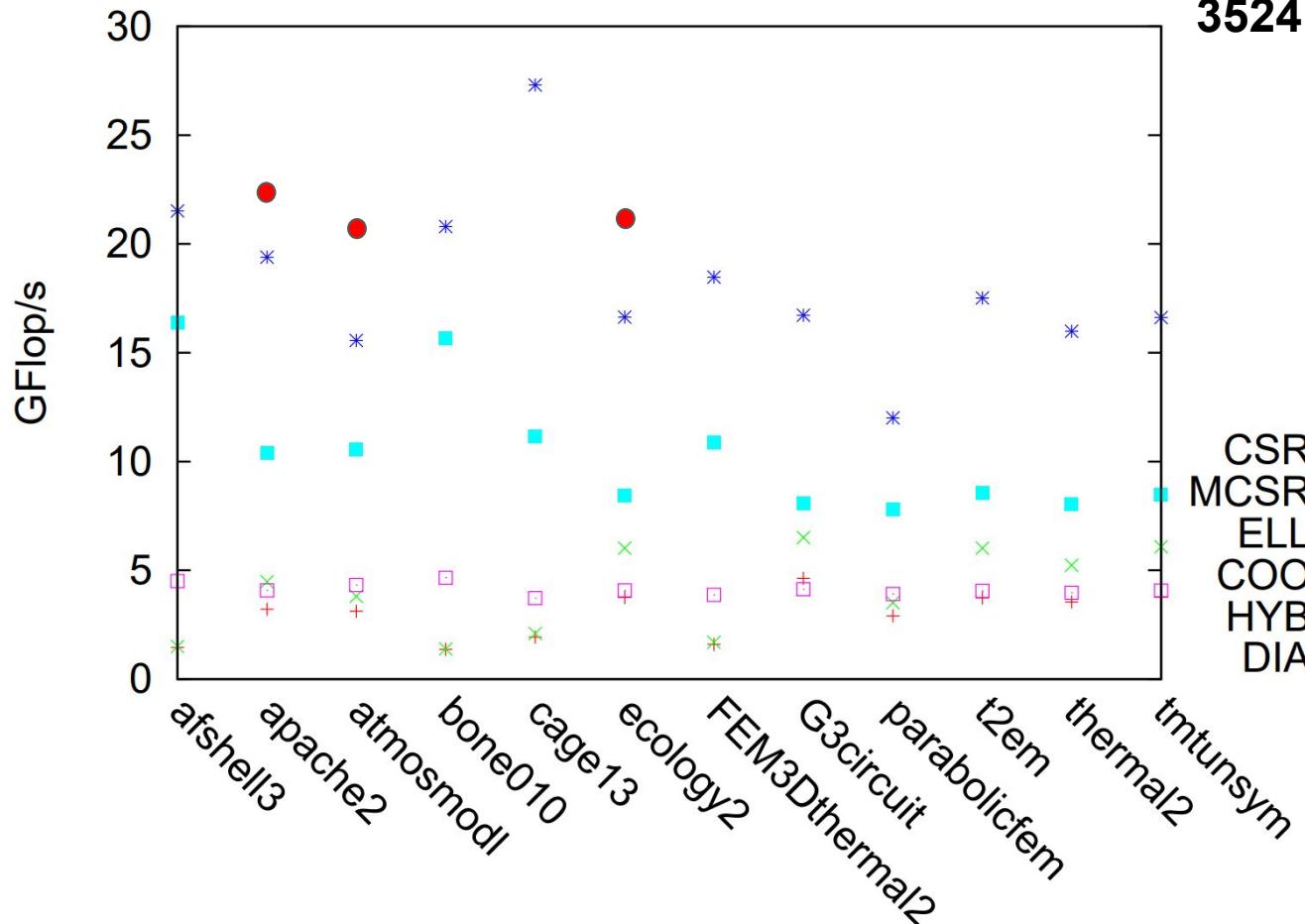
$A = B + C$ (для выбросов)

$Axv = Bxv + Cxv$

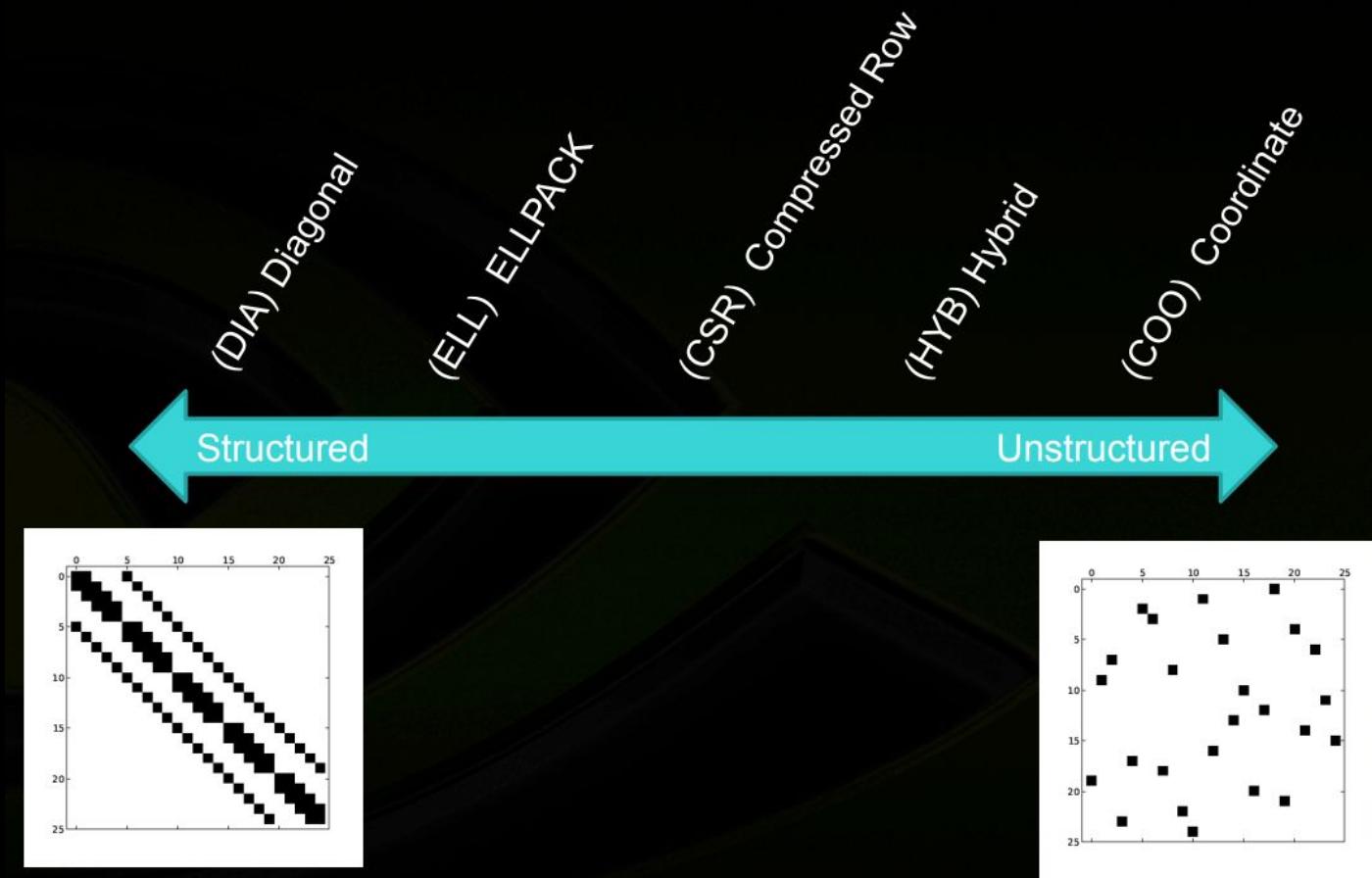


Sparse Matrix-Vector Multiplication on K20 GPU

208 Gb/s bandwidth
3524 GFlop/s

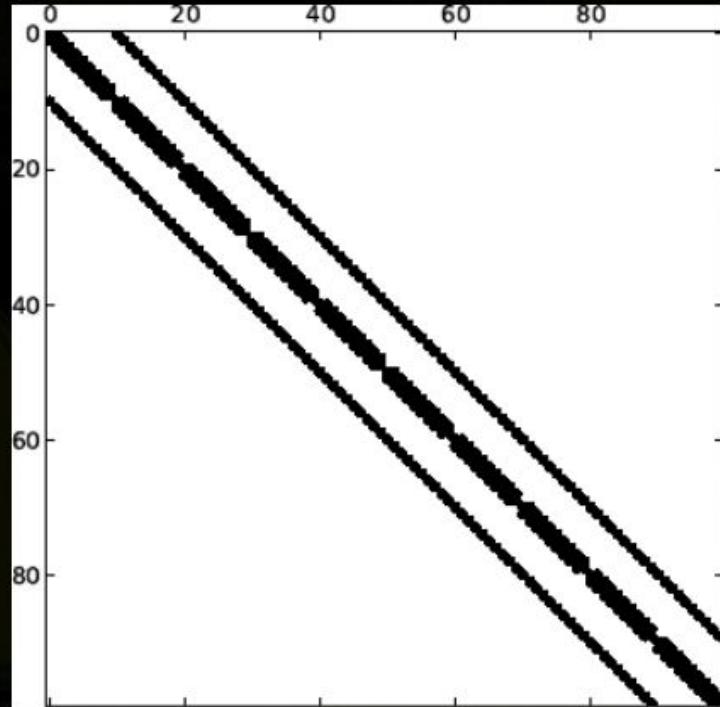


Источник: [Sparse Matrix-Vector Multiplication and Matrix Formats](#)



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

Structured Mesh



Matrix

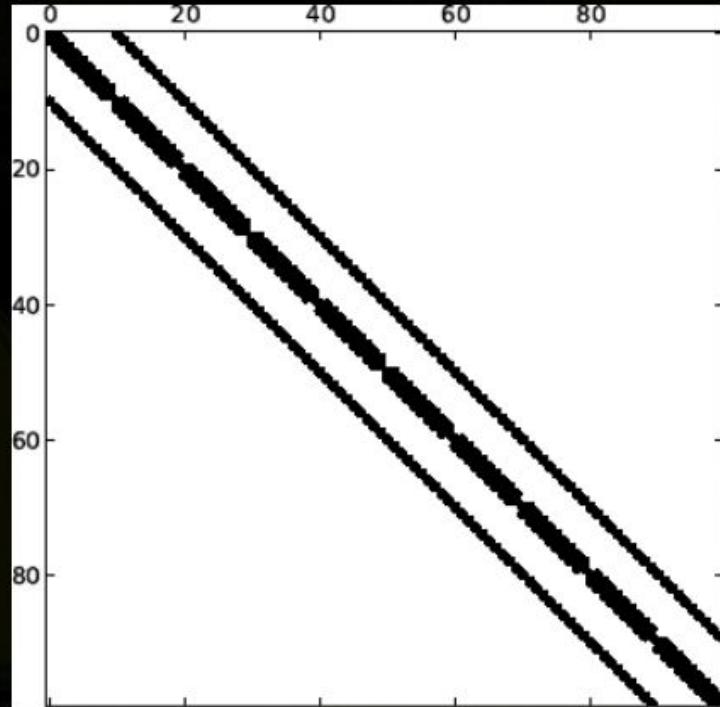
Format	float	double
COO		
CSR		
DIA		
ELL		
HYB		

Кто компактнее?

Bytes per Nonzero Entry

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

Structured Mesh



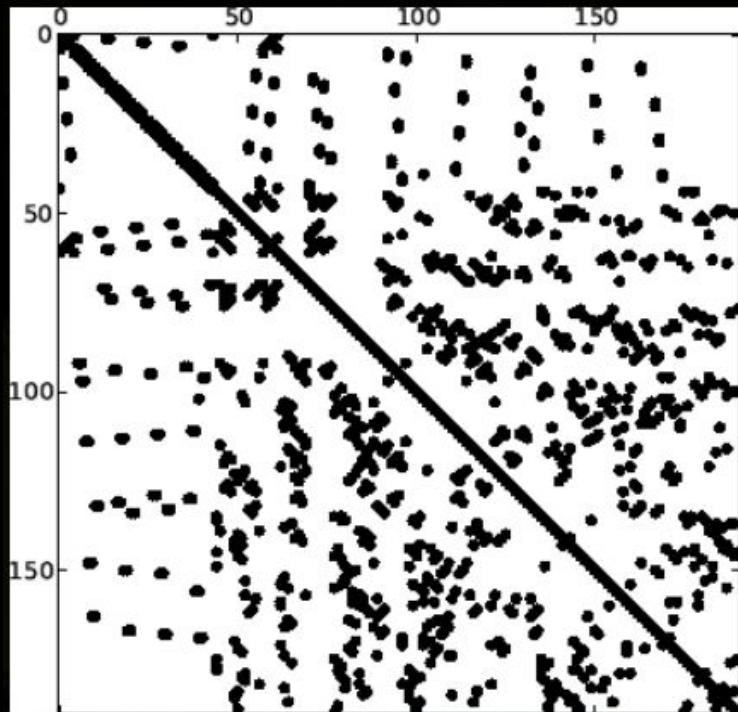
Matrix

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

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

Unstructured Mesh



Matrix

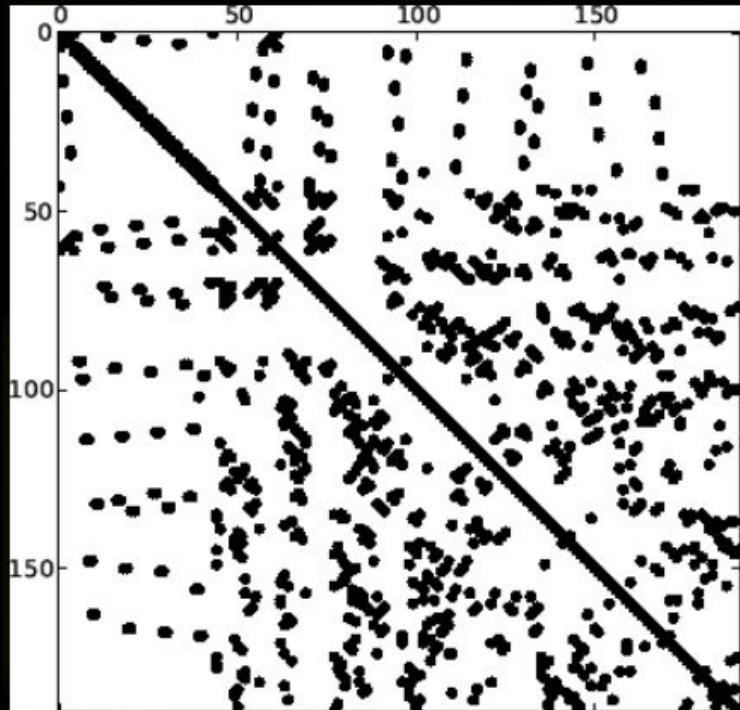
Bytes per Nonzero Entry

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

Format	float	double
COO		
CSR		
DIA		
ELL		
HYB		

Кто компактнее?

Unstructured Mesh

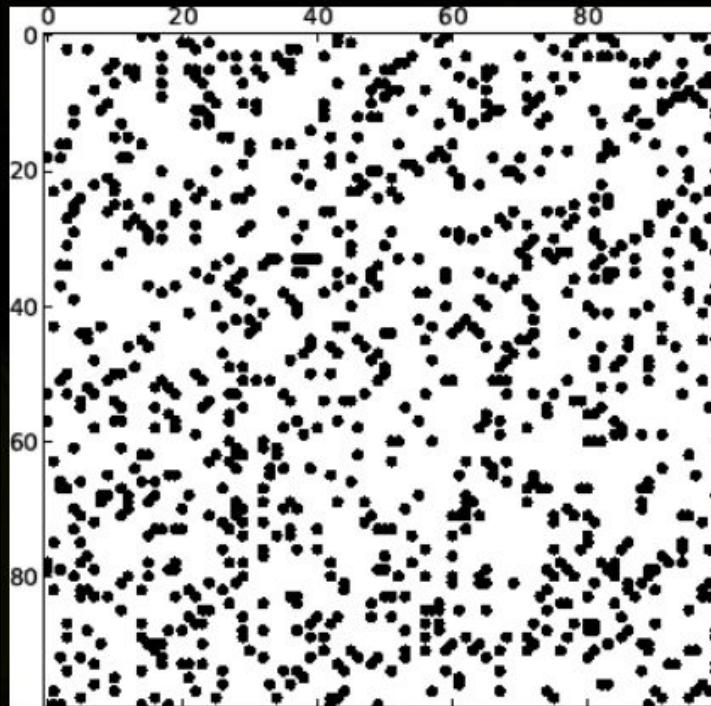


Matrix

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

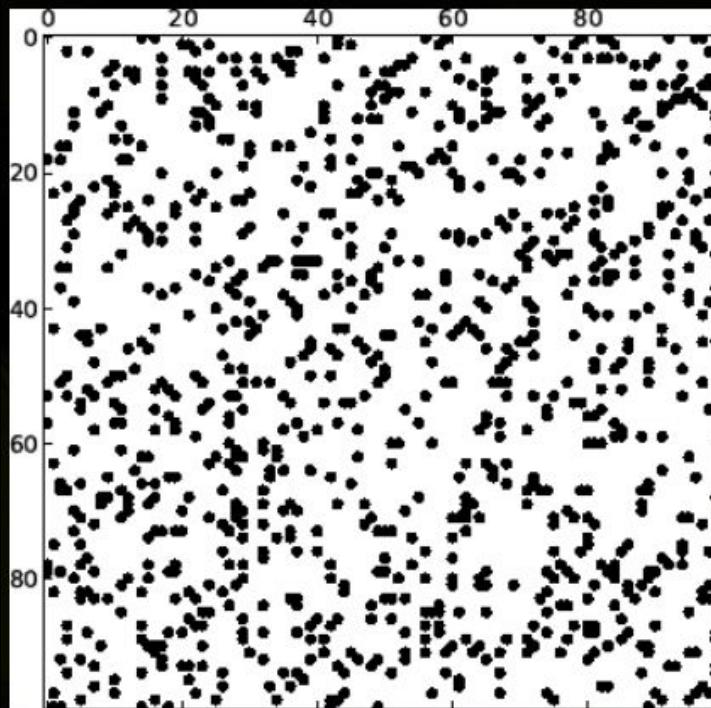
Источник: [Sparse Matrix Representations & Iterative Solvers](#)

Format	Float	Double
COO		
CSR		
DIA		
ELL		
HYB		

Кто компактнее?

Bytes per Nonzero Entry

Random



Matrix

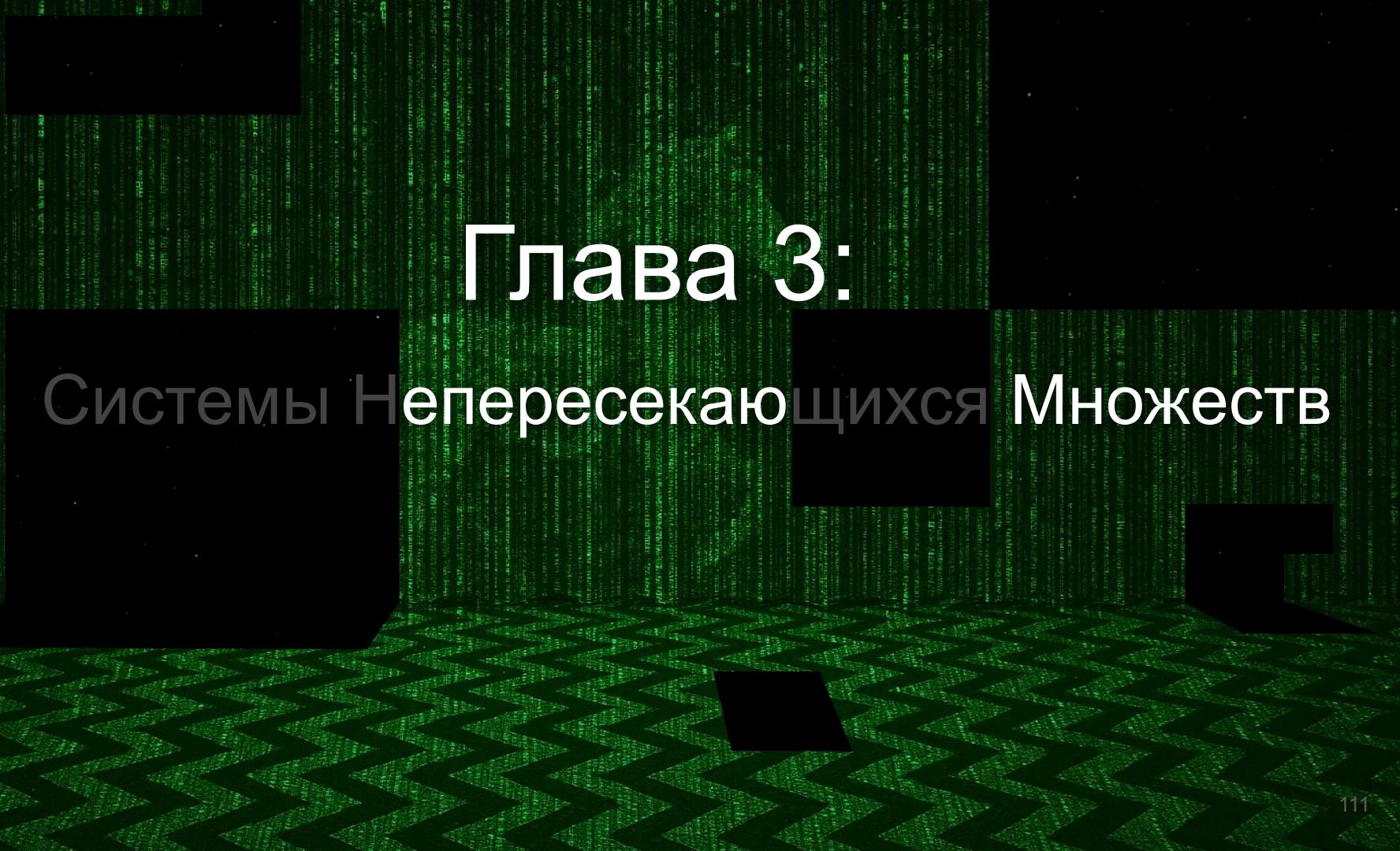
Bytes per Nonzero Entry

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

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

Системы Непересекающихся Множеств

Глава 3:



Система Непересекающихся Множеств - СНМ



Disjoint Set

Система Непересекающихся Множеств - СНМ



Disjoint Set

find(2) = 2

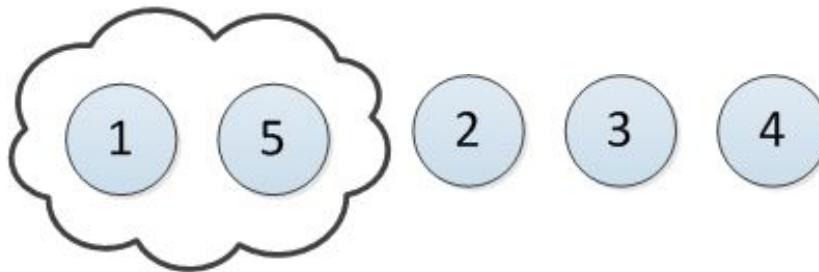
Система Непересекающихся Множеств - СНМ

Disjoint Set



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

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



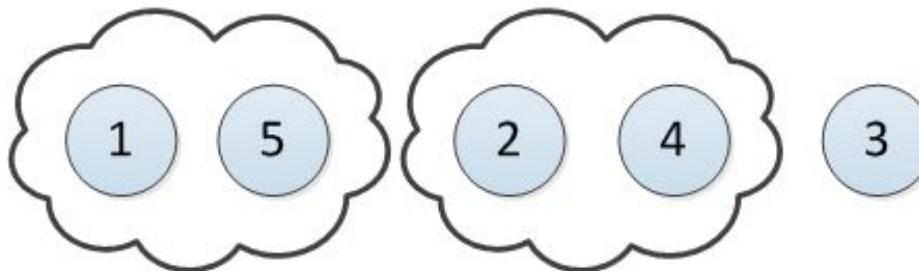
Система Непересекающихся Множеств - СНМ

Disjoint Set



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

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



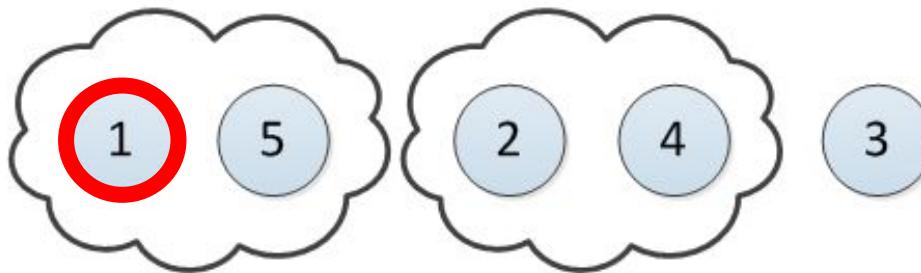
Система Непересекающихся Множеств - СНМ

Disjoint Set



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

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



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

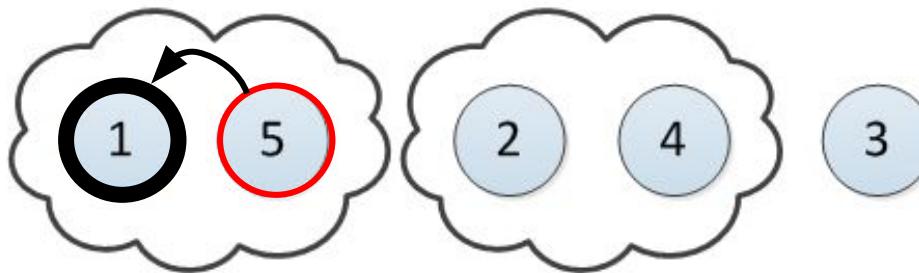
Система Непересекающихся Множеств - СНМ

Disjoint Set



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

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



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

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

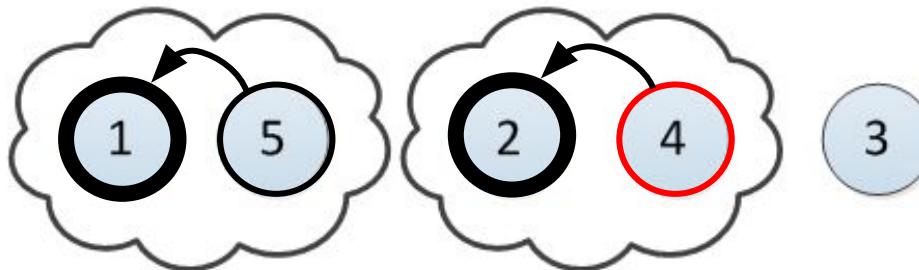
Система Непересекающихся Множеств - СНМ

Disjoint Set



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

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



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

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

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

[https://neerc.ifmo.ru/wiki/index.php?title=CHM_\(реализация_с_помощью_леса_корневых_деревьев\)](https://neerc.ifmo.ru/wiki/index.php?title=CHM_(реализация_с_помощью_леса_корневых_деревьев))

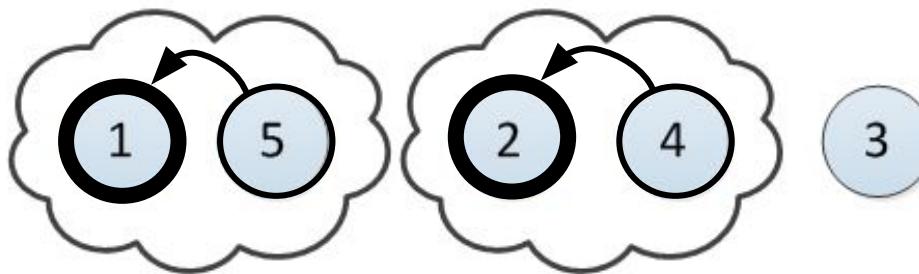
Система Непересекающихся Множеств - СНМ

Disjoint Set



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

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



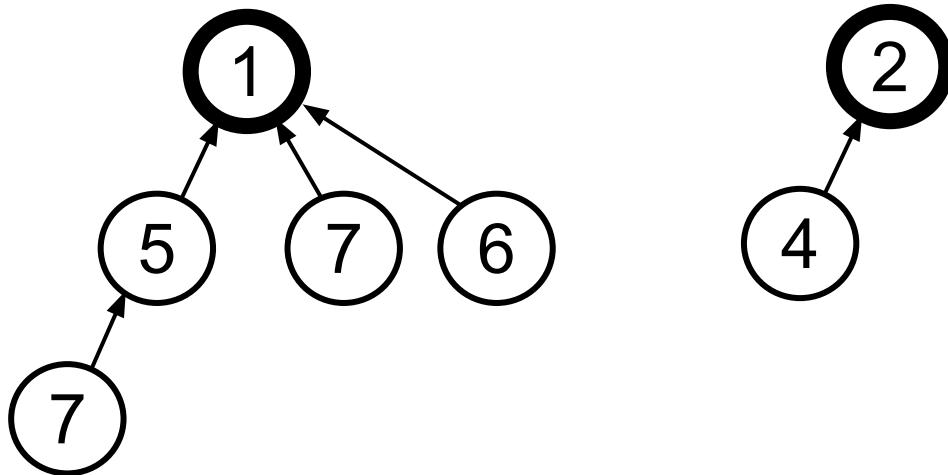
$\text{find}(1) = 1$
 $\text{find}(5) = 1$
 $\text{find}(4) = 2$

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

[https://neerc.ifmo.ru/wiki/index.php?title=СНМ_\(реализация_с_помощью_леса_корневых_деревьев\)](https://neerc.ifmo.ru/wiki/index.php?title=СНМ_(реализация_с_помощью_леса_корневых_деревьев))

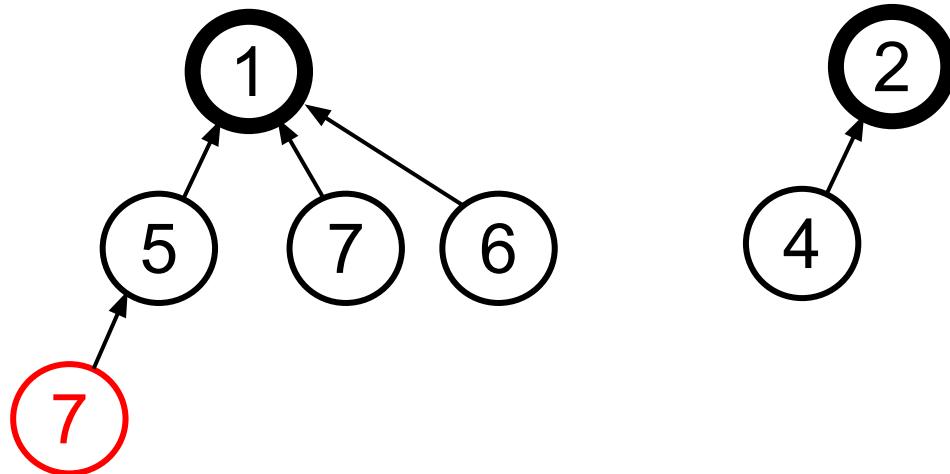
Система Непересекающихся Множеств - СНМ

Disjoint Set



Система Непересекающихся Множеств - СНМ

Disjoint Set

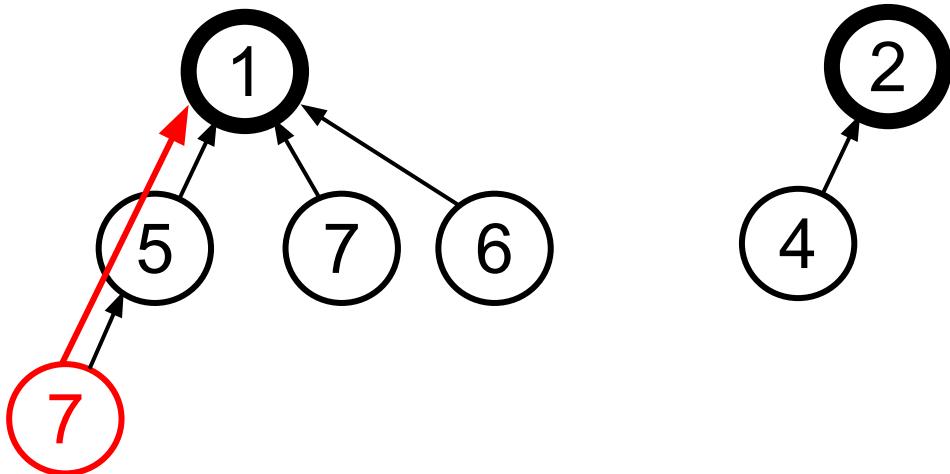


get(7) как выполнить?

[https://neerc.ifmo.ru/wiki/index.php?title=СНМ_\(реализация_с_помощью_леса_корневых_деревьев\)](https://neerc.ifmo.ru/wiki/index.php?title=СНМ_(реализация_с_помощью_леса_корневых_деревьев))

Система Непересекающихся Множеств - СНМ

Disjoint Set

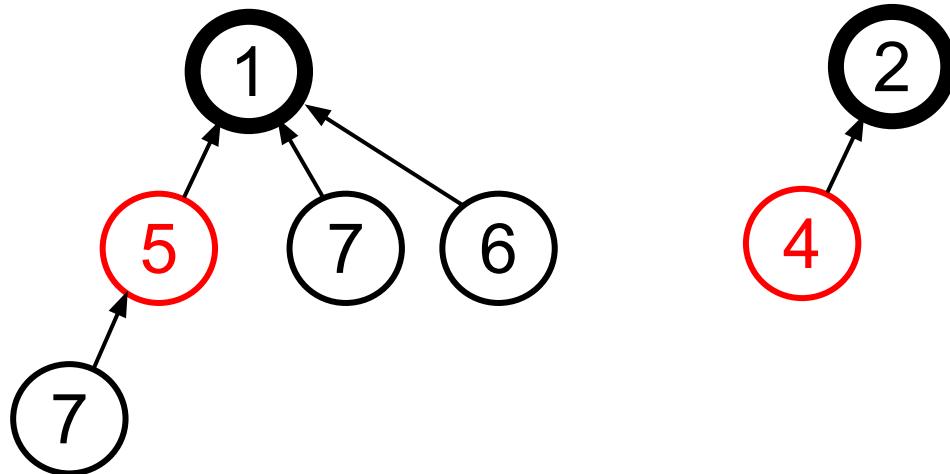


get(7) как выполнить?

[https://neerc.ifmo.ru/wiki/index.php?title=СНМ_\(реализация_с_помощью_леса_корневых_деревьев\)](https://neerc.ifmo.ru/wiki/index.php?title=СНМ_(реализация_с_помощью_леса_корневых_деревьев))

Система Непересекающихся Множеств - СНМ

Disjoint Set

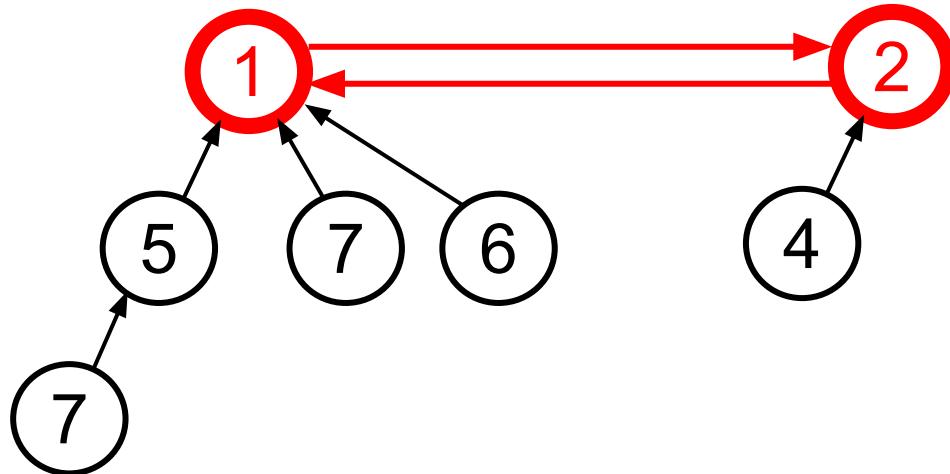


$\text{union}(5, 4)$ как выполнить?

[https://neerc.ifmo.ru/wiki/index.php?title=CHM_\(реализация_с_помощью_леса_корневых_деревьев\)](https://neerc.ifmo.ru/wiki/index.php?title=CHM_(реализация_с_помощью_леса_корневых_деревьев))

Система Непересекающихся Множеств - СНМ

Disjoint Set

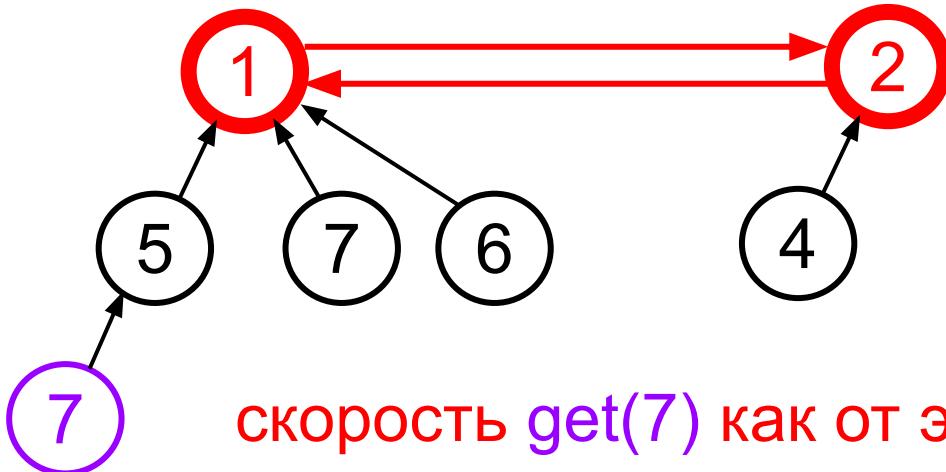


$\text{union}(1, 2)$ как выполнить?

[https://neerc.ifmo.ru/wiki/index.php?title=CHM_\(реализация_с_помощью_леса_корневых_деревьев\)](https://neerc.ifmo.ru/wiki/index.php?title=CHM_(реализация_с_помощью_леса_корневых_деревьев))

Система Непересекающихся Множеств - СНМ

Disjoint Set

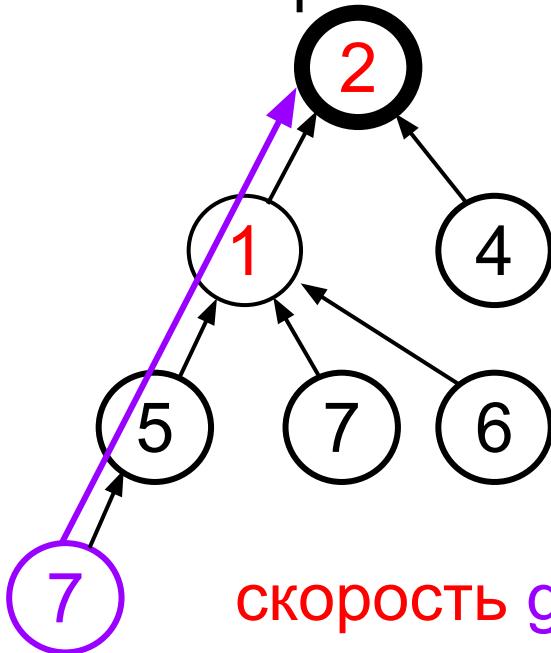


скорость $get(7)$ как от этого зависит?

$union(1, 2)$ как выполнить?

Система Непересекающихся Множеств - СНМ

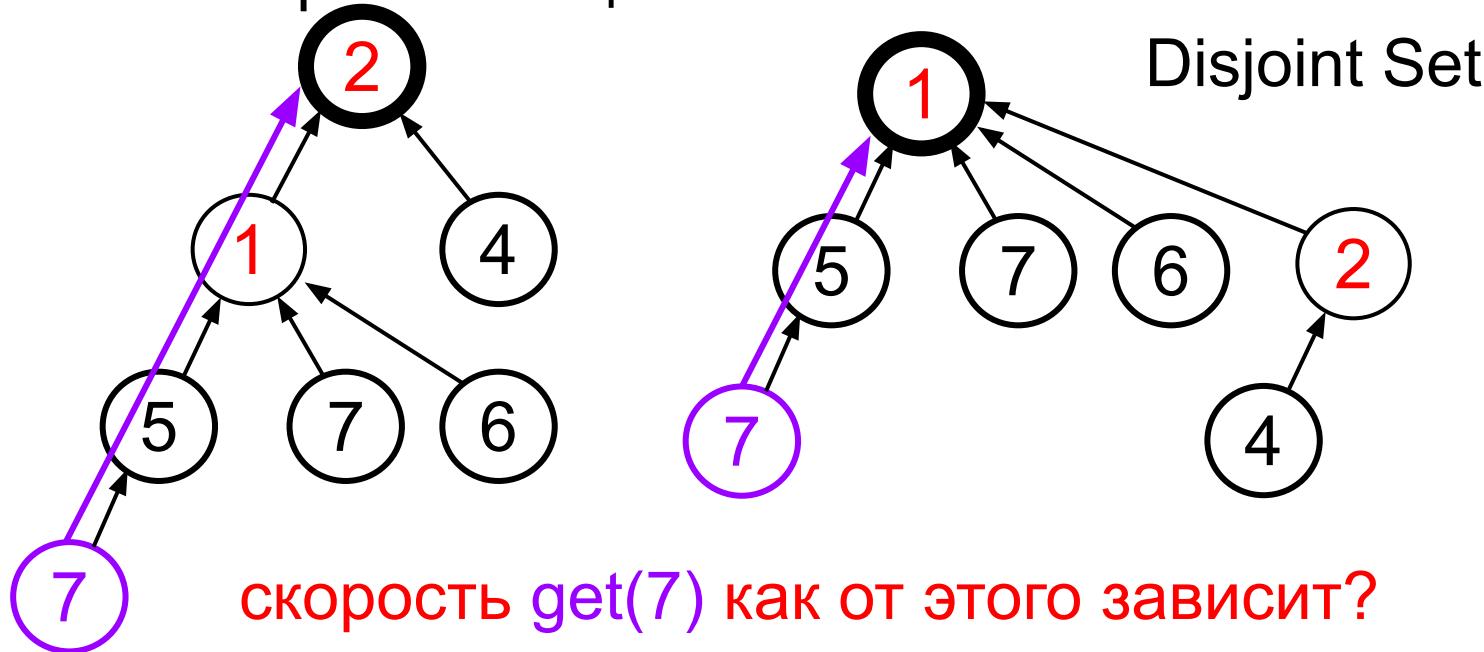
Disjoint Set



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

$union(1, 2)$ как выполнить?

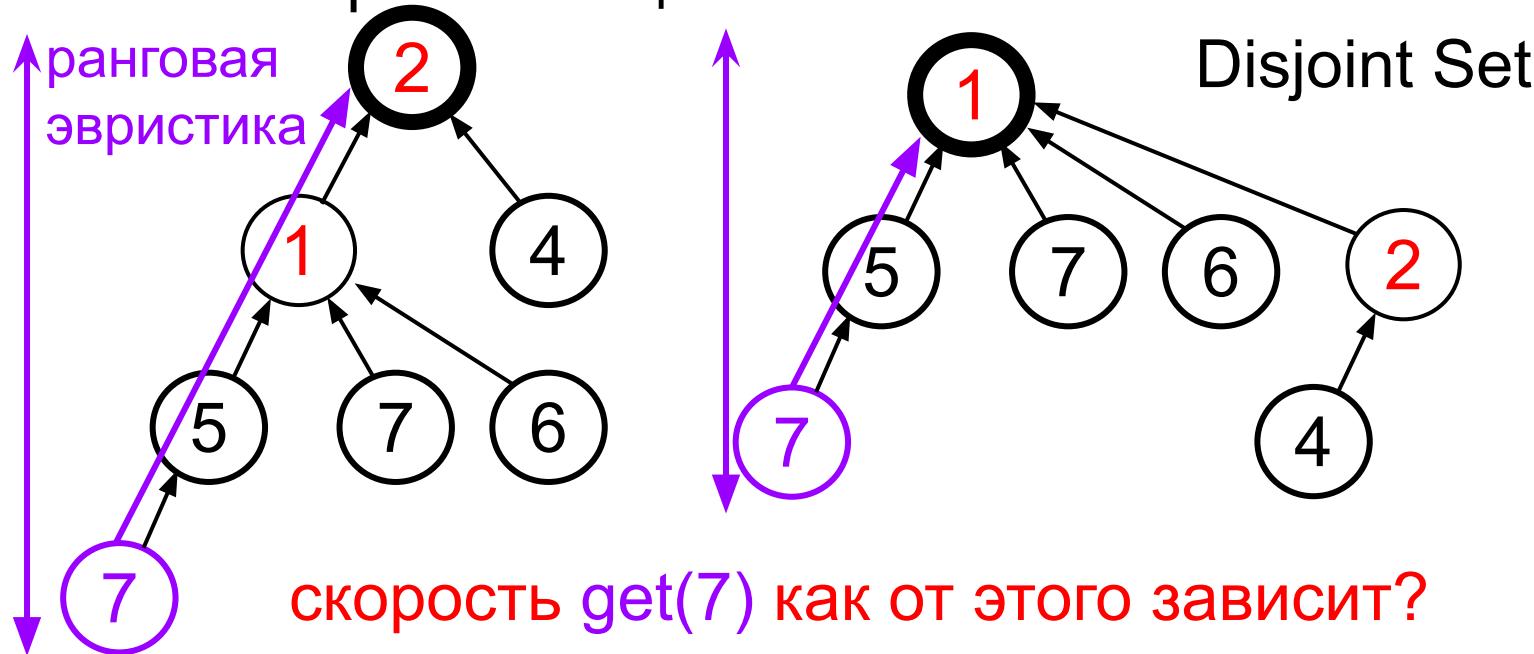
Система Непересекающихся Множеств - СНМ



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

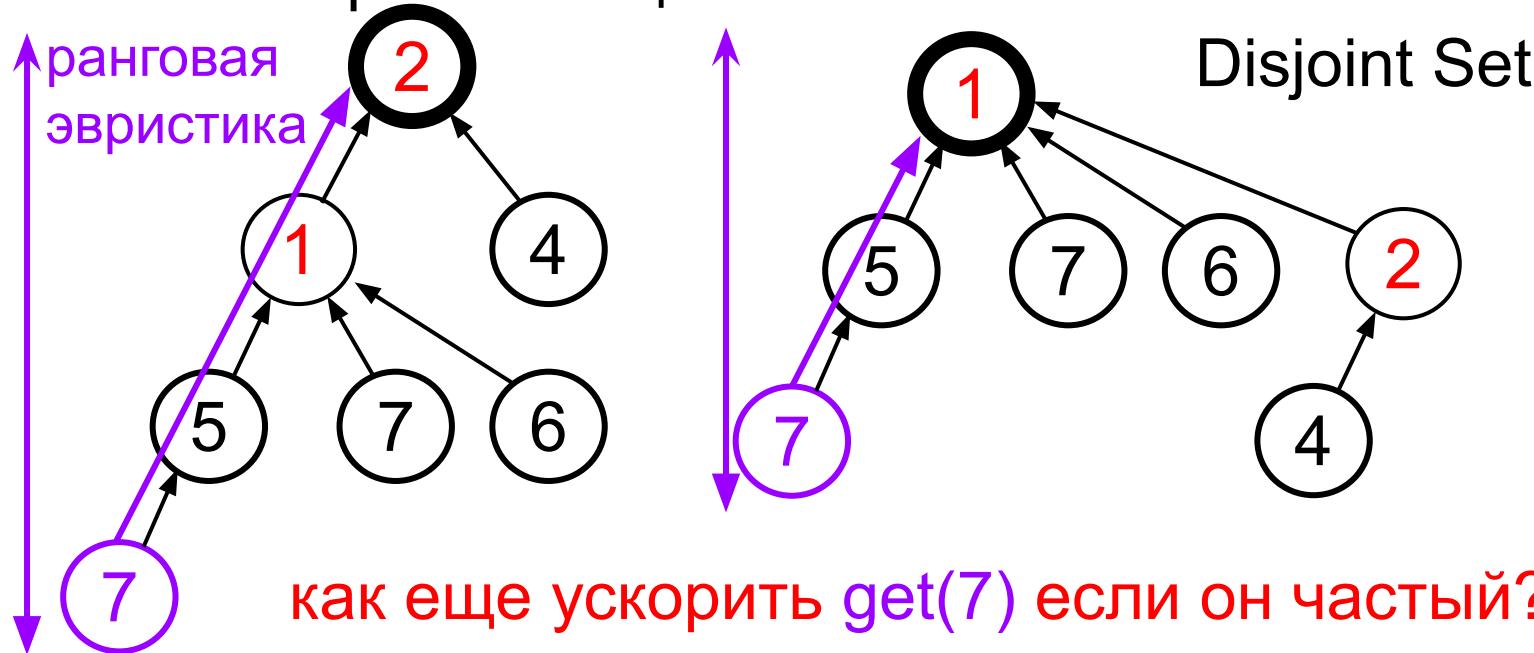
$union(1, 2)$ как выполнить?

Система Непересекающихся Множеств - СНМ

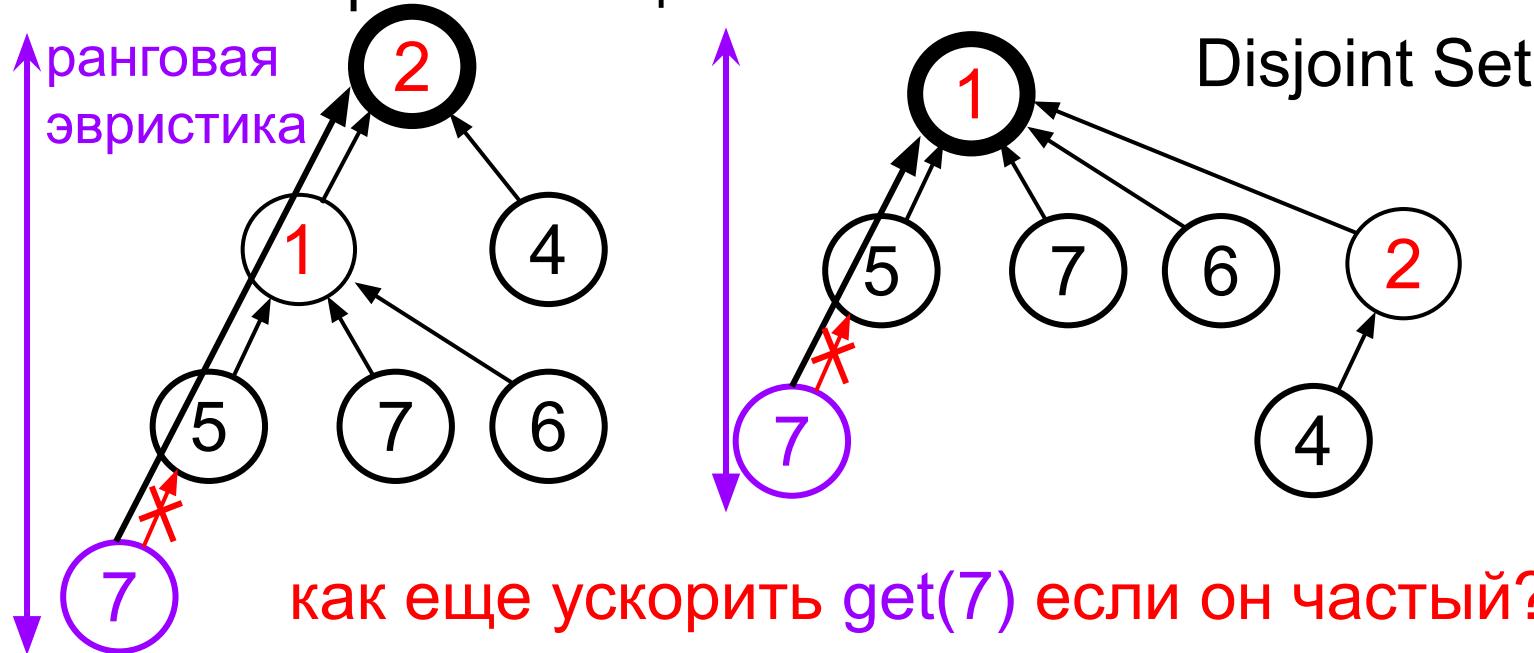


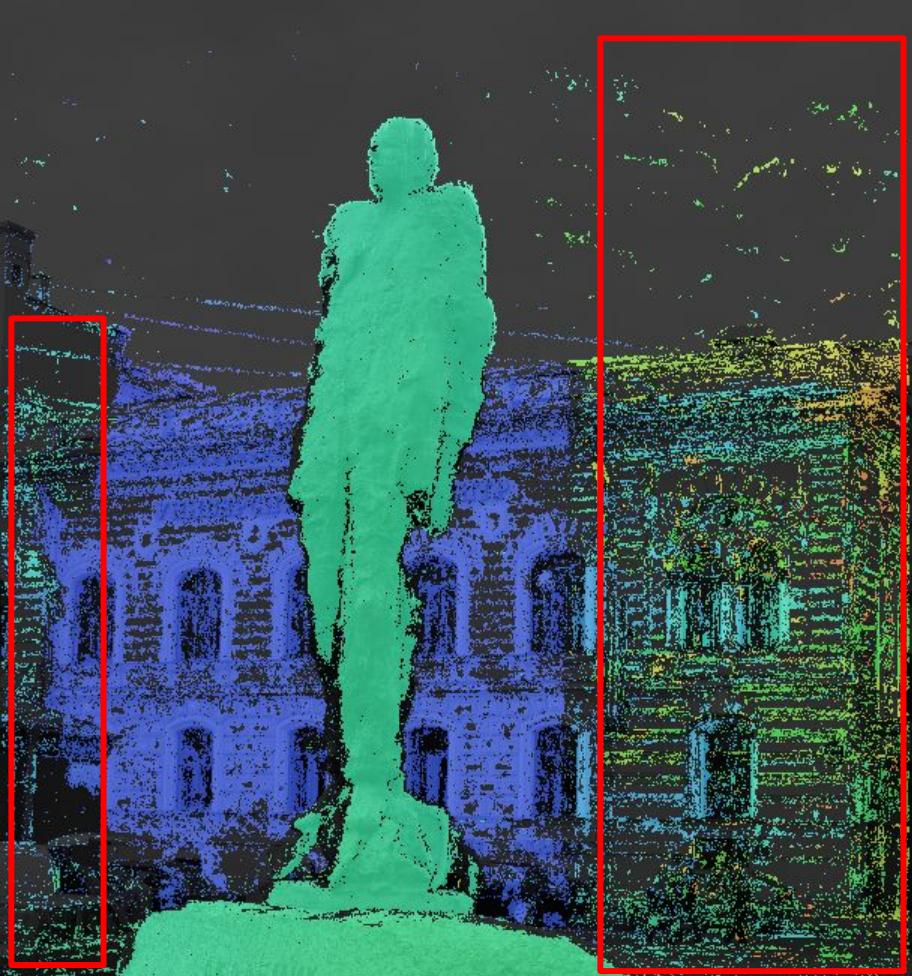
[https://neerc.ifmo.ru/wiki/index.php?title=CHM_\(реализация_с_помощью_леса_корневых_деревьев\)](https://neerc.ifmo.ru/wiki/index.php?title=CHM_(реализация_с_помощью_леса_корневых_деревьев))

Система Непересекающихся Множеств - СНМ

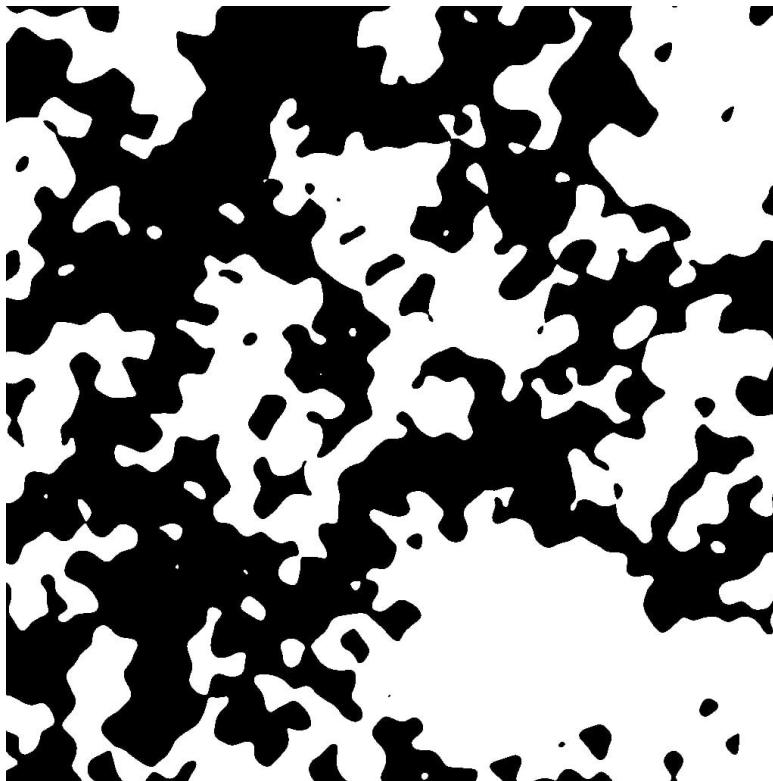


Система Непересекающихся Множеств - СНМ





Система Непересекающихся Множеств - СНМ



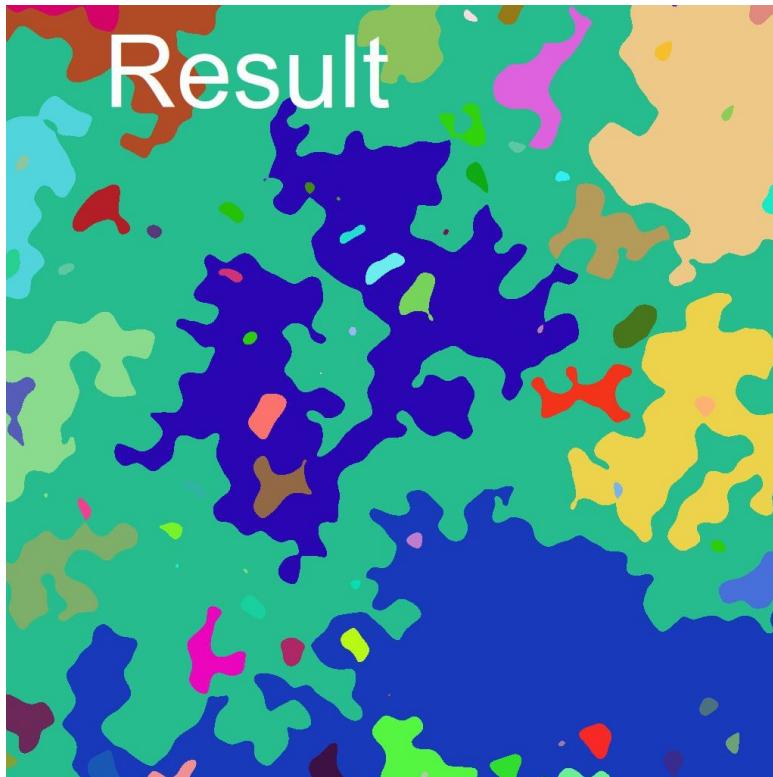
Disjoint Set

Как найти площадь каждого острова?

(например для speckles filtering -
фильтрации выбросов)

Система Непересекающихся Множеств - СНМ

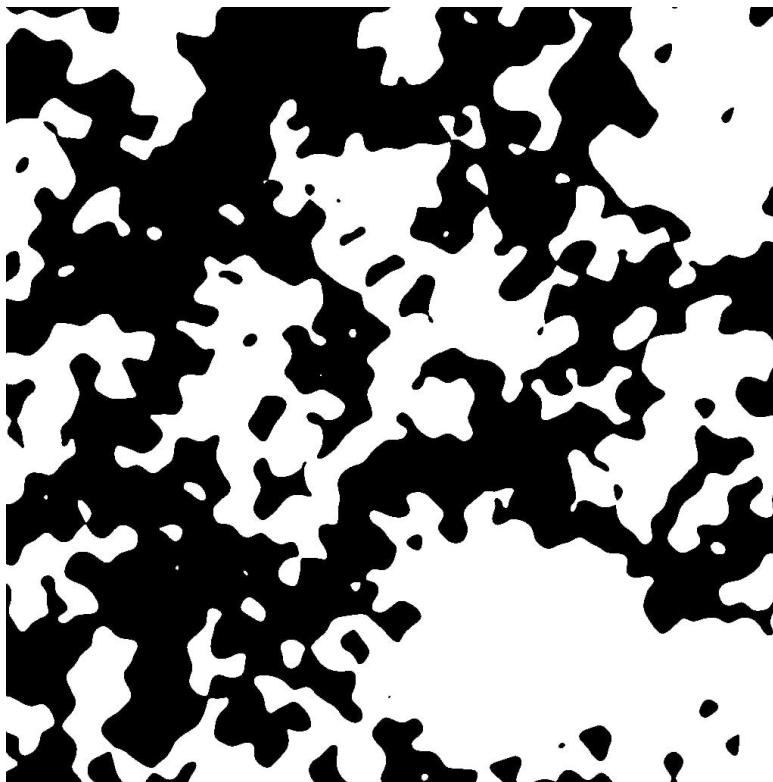
Disjoint Set



Как найти площадь каждого острова?

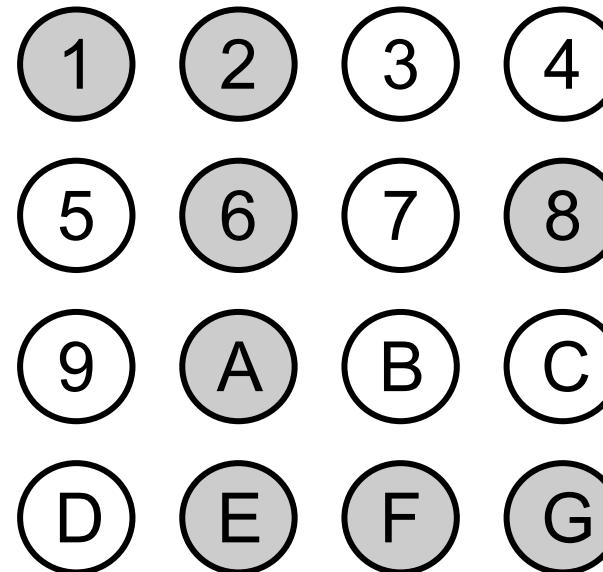
(например для speckles filtering -
фильтрации выбросов)

Система Непересекающихся Множеств - СНМ



Disjoint Set

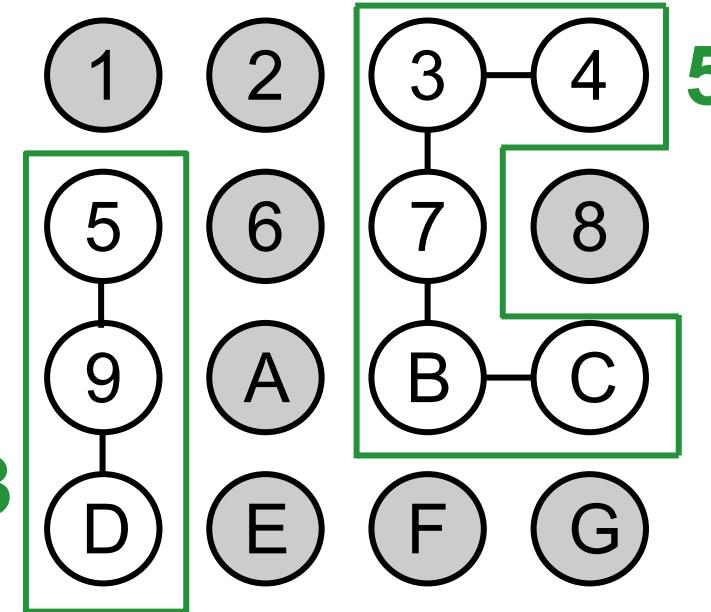
Как найти площадь каждого острова?



Система Непересекающихся Множеств - СНМ

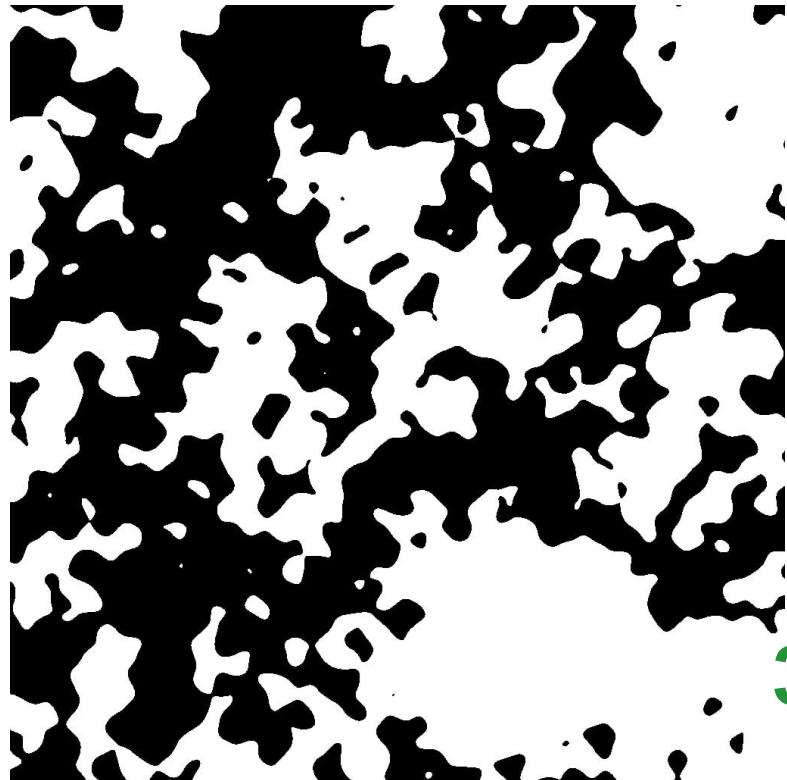


Как найти площадь каждого острова?

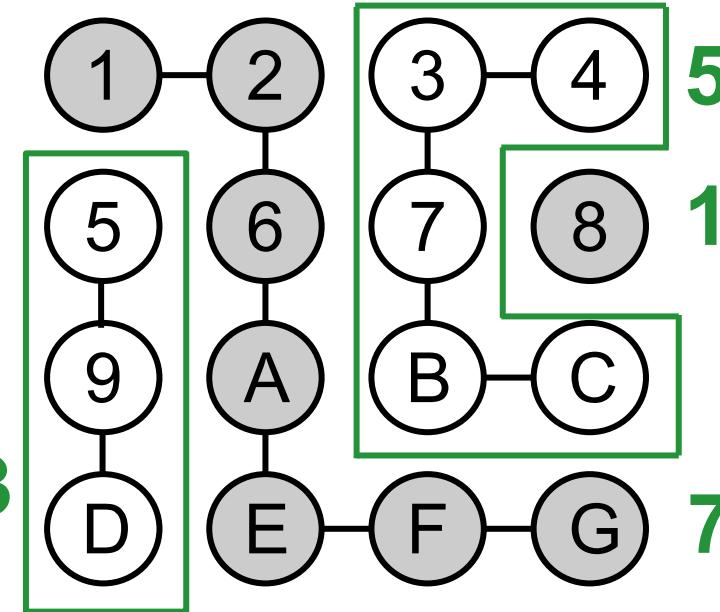


Система Непересекающихся Множеств - СНМ

Disjoint Set



Как найти площадь каждого острова?

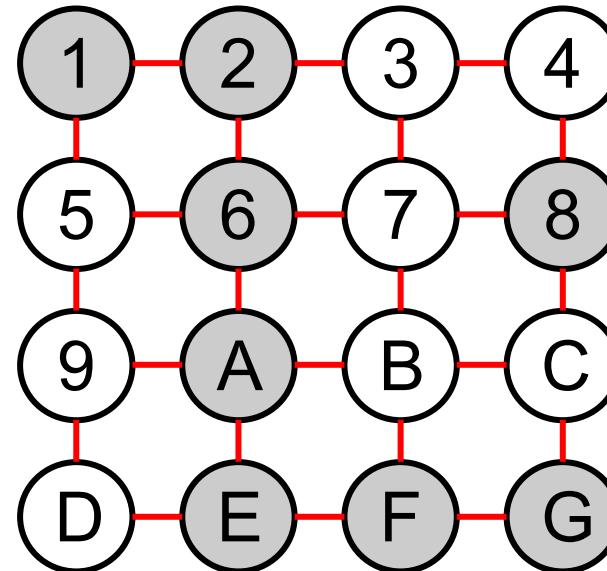


Система Непересекающихся Множеств - СНМ

Disjoint Set



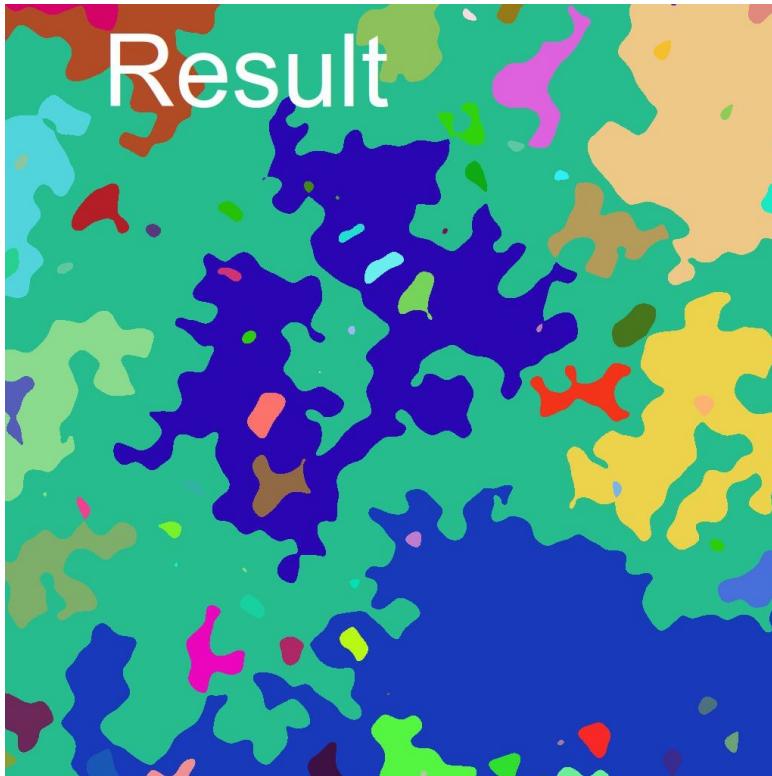
Как найти площадь каждого острова?



Можем ли
выполнять все
union операции
параллельно?

Система Непересекающихся Множеств - СНМ

Disjoint Set



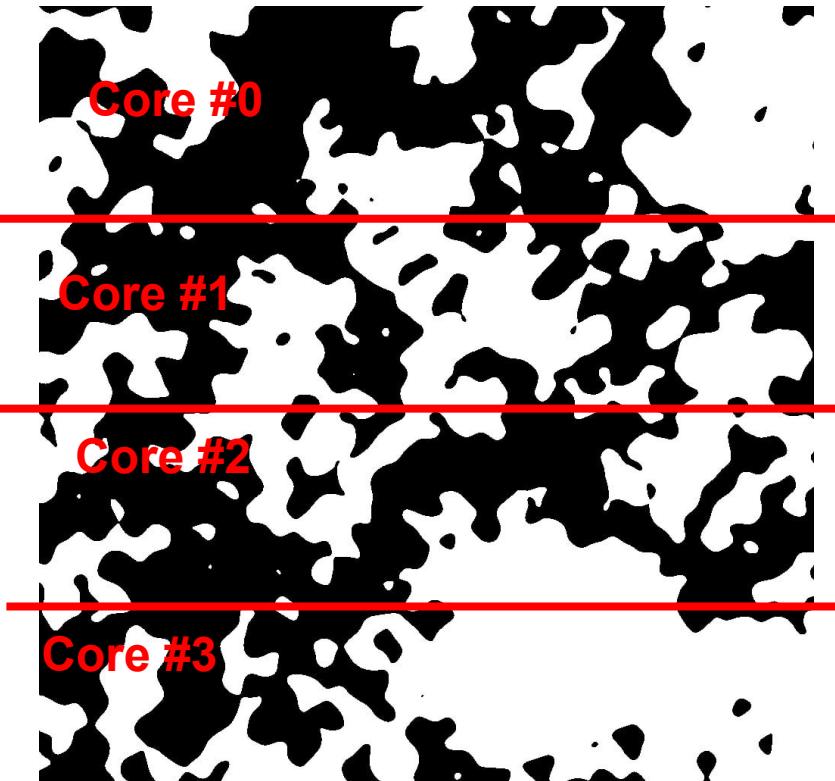
Как найти площадь каждого острова?

(например для speckles filtering -
фильтрации выбросов)

Как распараллелить на CPU?

Система Непересекающихся Множеств - СНМ

Disjoint Set



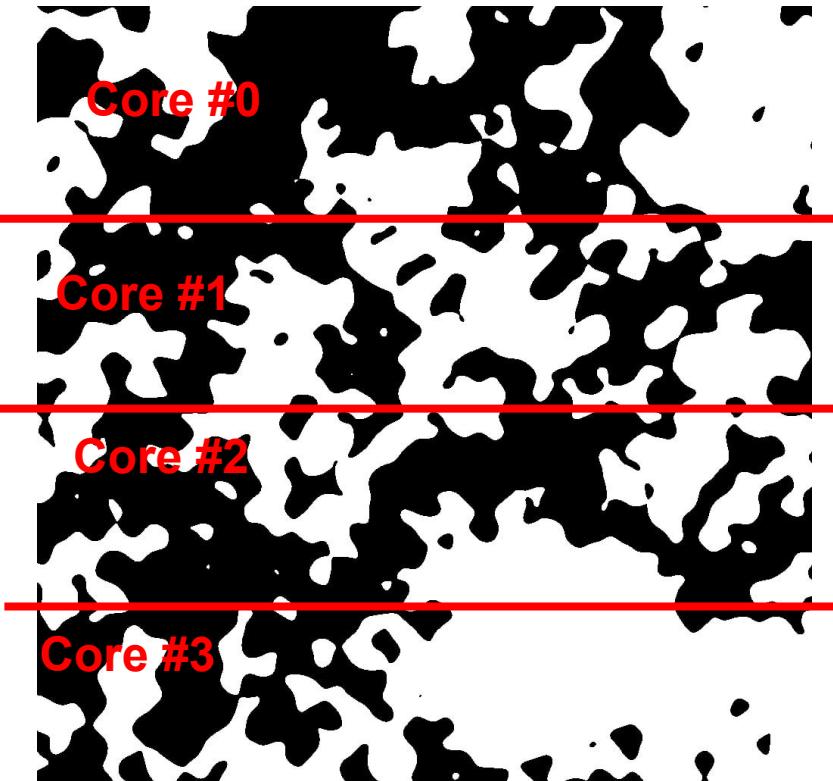
Как найти площадь каждого острова?

(например для speckles filtering - фильтрации выбросов)

Как распараллелить на CPU?

Система Непересекающихся Множеств - СНМ

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 - СНМ



Disjoint Set

Как найти площадь каждого острова?

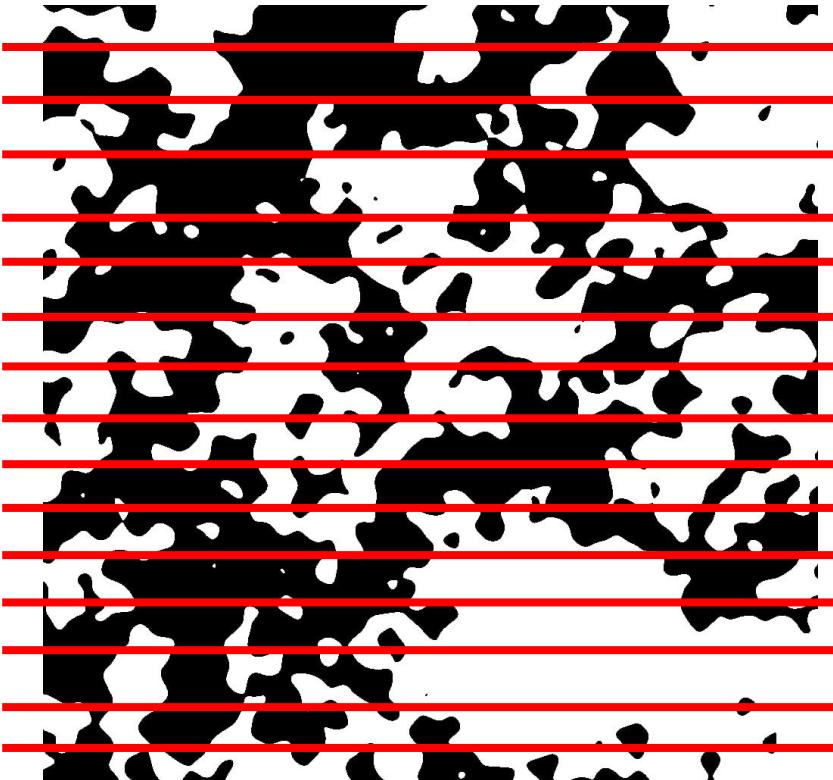
(например для speckles filtering - фильтрации выбросов)

Как распараллелить на CPU?

Как распараллелить на GPU?

Система Непересекающихся Множеств - СНМ

Disjoint Set



Как найти площадь каждого острова?

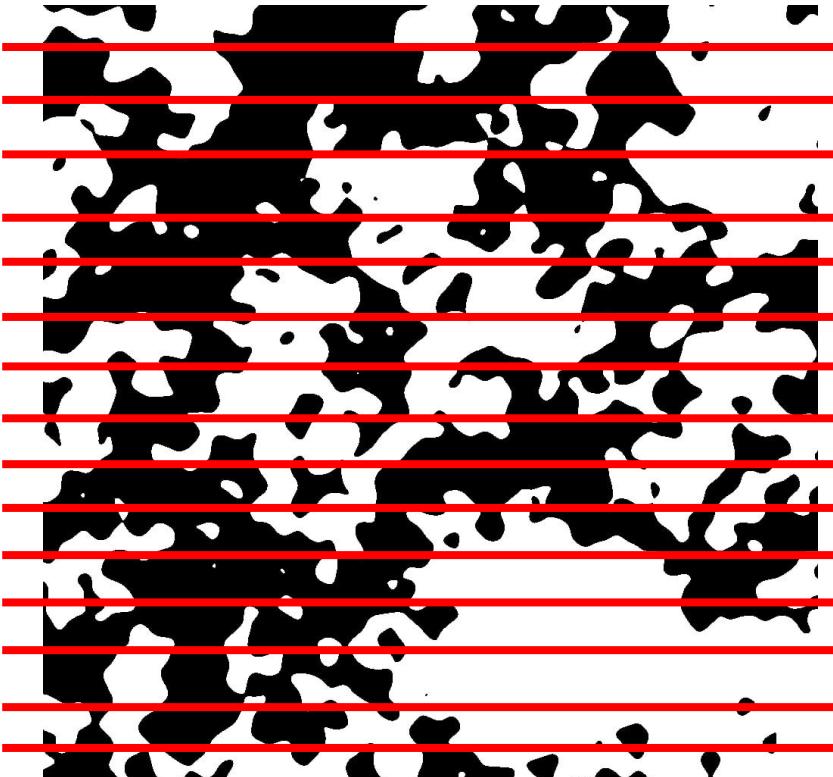
(например для speckles filtering -
фильтрации выбросов)

Как распараллелить на CPU?

Как распараллелить на GPU?

Система Непересекающихся Множеств - СНМ

Disjoint Set



Как найти площадь каждого острова?

(например для speckles filtering -
фильтрации выбросов)

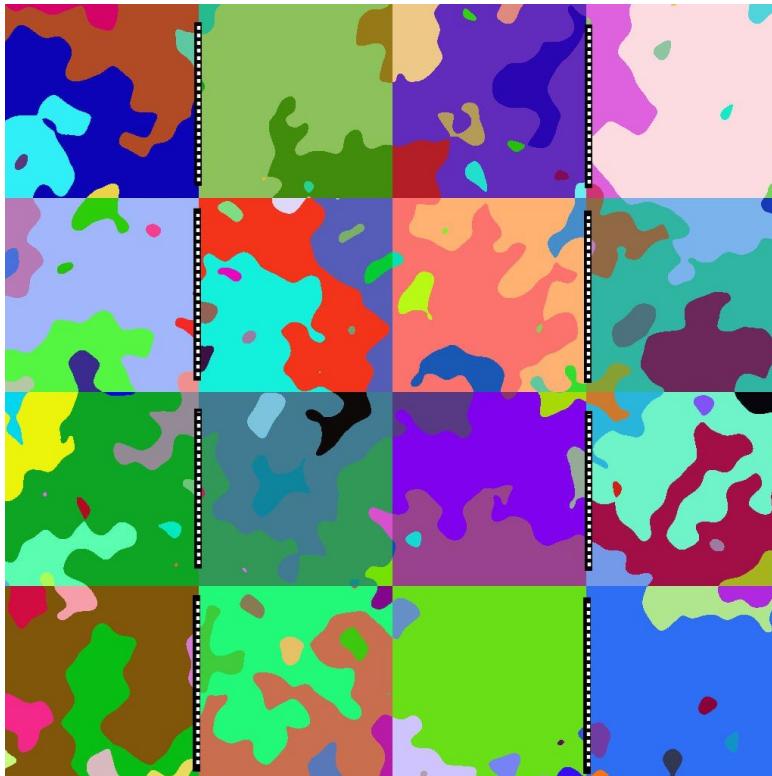
Как распараллелить на CPU?

**Как распараллелить на GPU?
А как потом доделать работу на швах?**

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

Система Непересекающихся Множеств - СНМ

Disjoint Set



Как найти площадь каждого острова?

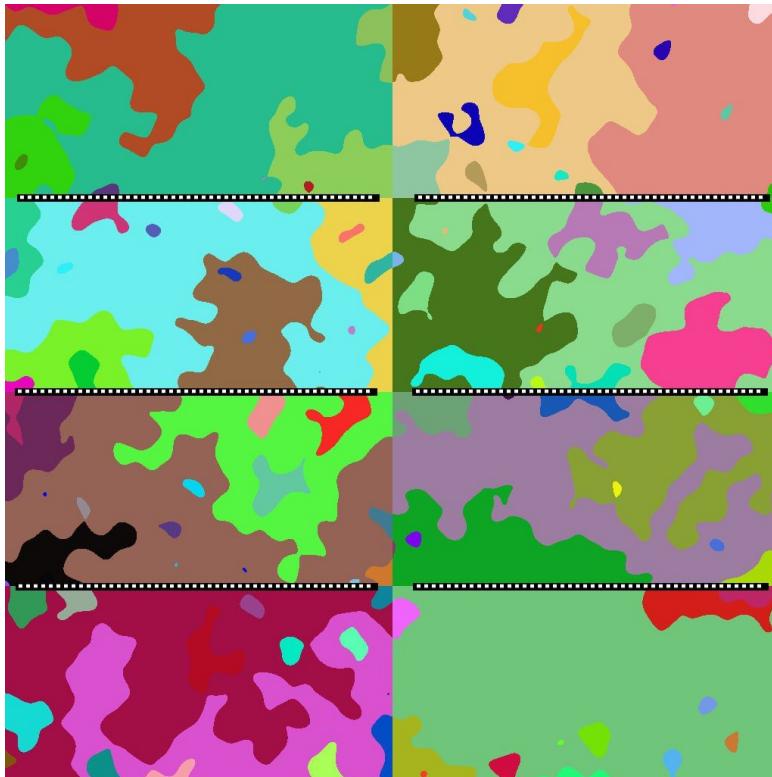
(например для speckles filtering -
фильтрации выбросов)

Как распараллелить на CPU?

Как распараллелить на GPU?

Система Непересекающихся Множеств - СНМ

Disjoint Set



Как найти площадь каждого острова?

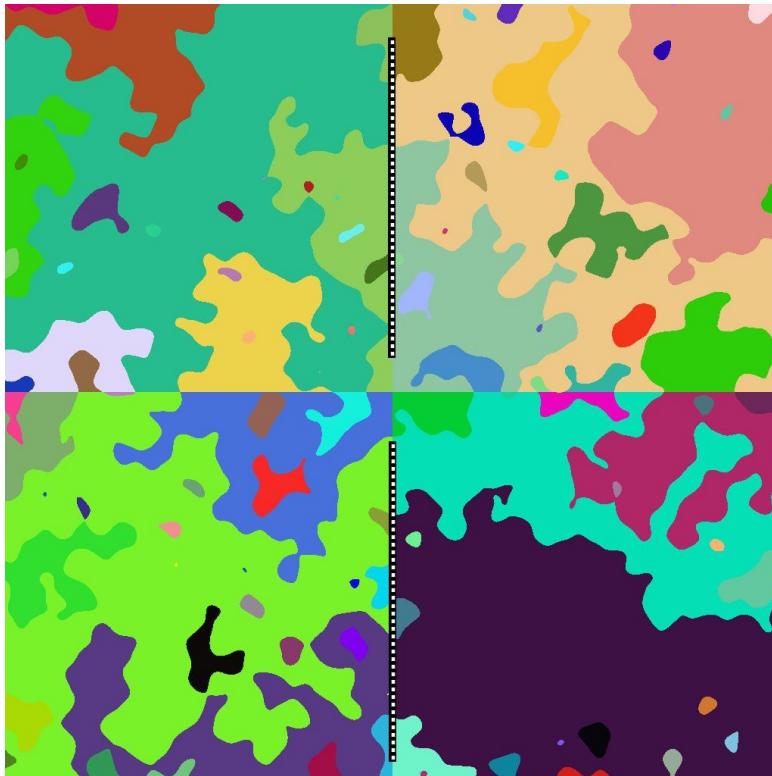
(например для speckles filtering -
фильтрации выбросов)

Как распараллелить на CPU?

Как распараллелить на GPU?

Система Непересекающихся Множеств - СНМ

Disjoint Set



Как найти площадь каждого острова?

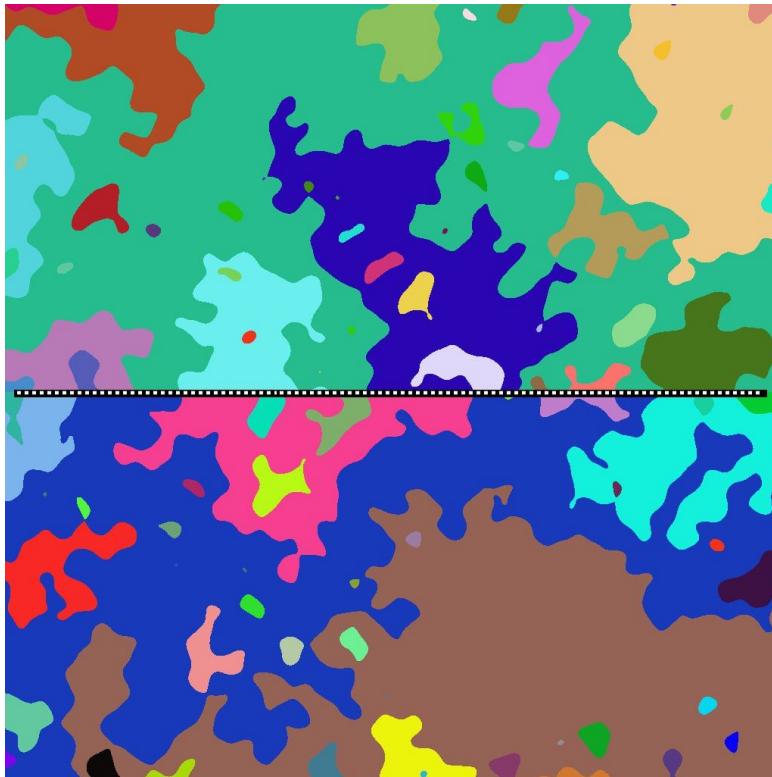
(например для speckles filtering -
фильтрации выбросов)

Как распараллелить на CPU?

Как распараллелить на GPU?

Система Непересекающихся Множеств - СНМ

Disjoint Set



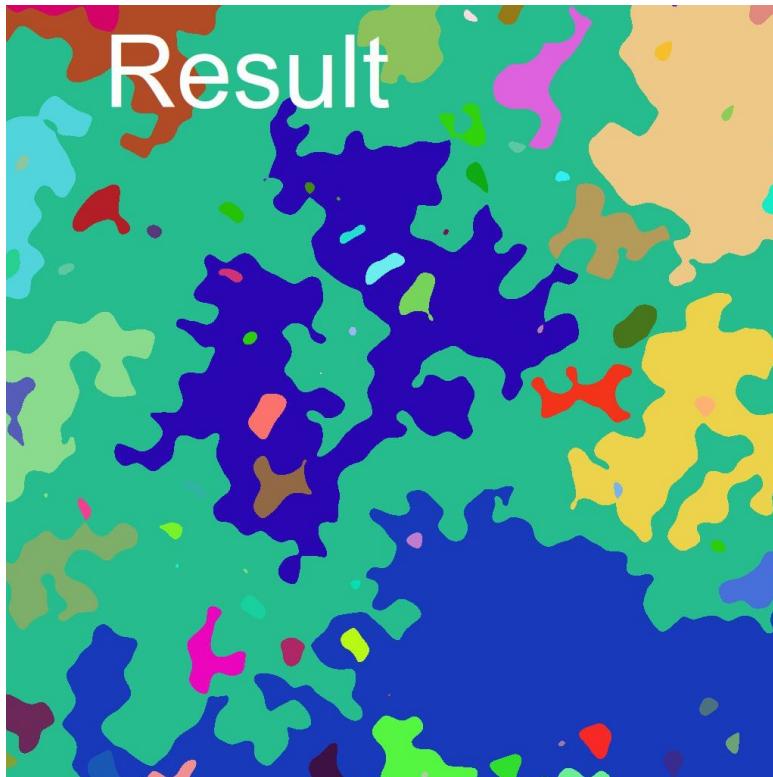
Как найти площадь каждого острова?

(например для speckles filtering -
фильтрации выбросов)

Как распараллелить на CPU?

Как распараллелить на GPU?

Система Непересекающихся Множеств - СНМ



Disjoint Set

Как найти площадь каждого острова?

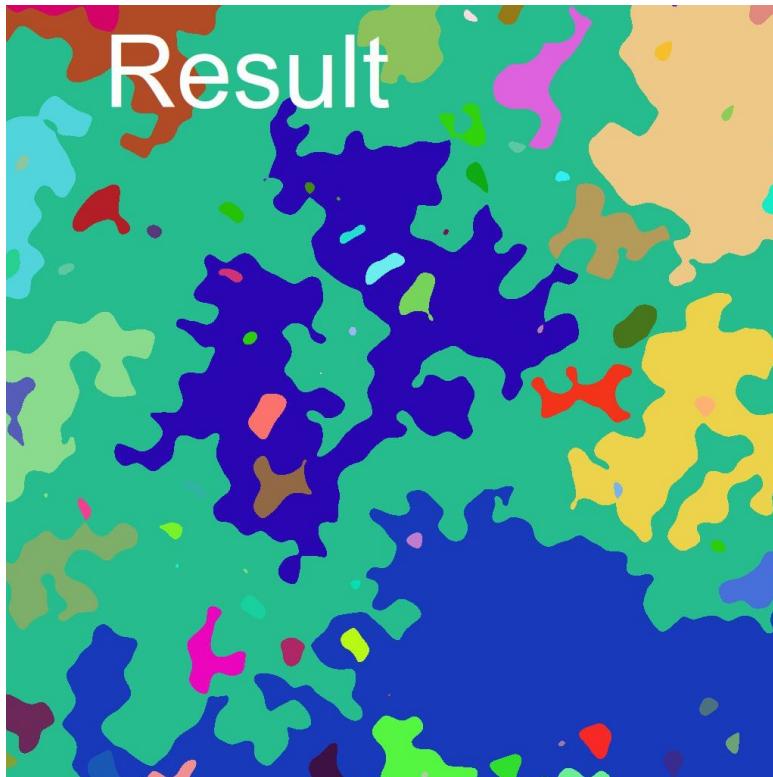
(например для speckles filtering - фильтрации выбросов)

Как распараллелить на CPU?

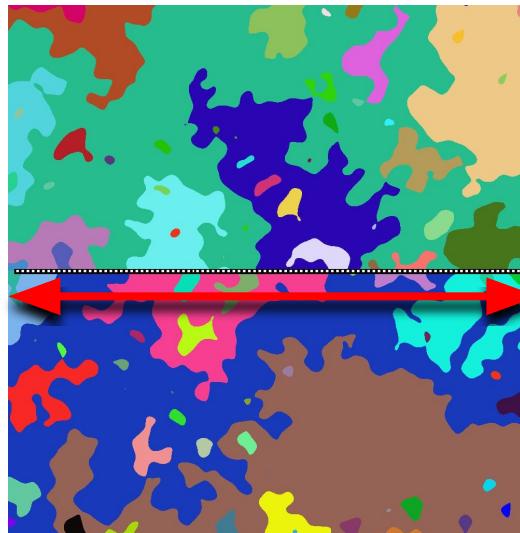
Как распараллелить на GPU?

Какой из запусков кернела самый медленный?

Система Непересекающихся Множеств - СНМ



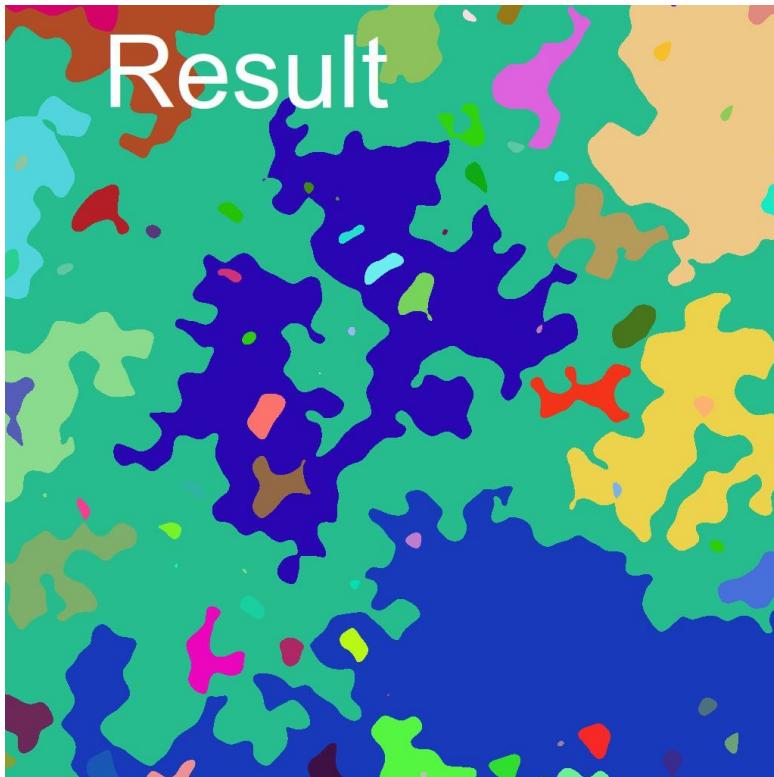
Result



Disjoint Set

Какой из запусков кернела самый медленный?

Система Непересекающихся Множеств - СНМ



Disjoint Set

Картинка: **4096x4096**
Множеств: **91**
CPU 5960X: **2.8 sec**
CPU 5960X OpenMP: **0.73 sec**
GPU RTX 3080 Vulkan: **0.094 sec**



Вопросы?



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