Параллельная обработка больших графов

Занятие 6

А.С. Семенов

dislab.org

Community Detection

- Задача Community Detection: поиск сообществ в неориентированном графе с весами
- неориентированный граф G = (V, E, W)
- V – множество вершин
- E ⊂ V⨯V – множество ребер
- W: E → R[0;1] – весовая функция
- Цель задачи — найти такое разбиение графа G на множество C непересекающихся сообществ вершин $c_i \subseteq V$:

$$\cup c_i = V, \forall c_i \in C$$

$$c_i \cap c_j = \emptyset, \forall c_i, c_j \in C, i \neq j,$$

Постановка задачи

Необходимо максимизировать функционал модулярности Q:

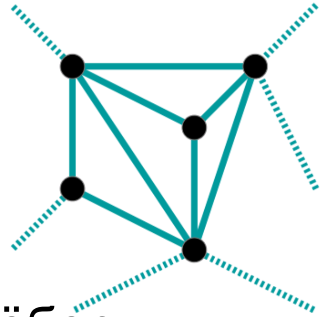
$$Q = \sum_{c \in C} \left(\frac{\sum_{in}^{c}}{m} - \left[\frac{\sum_{tot}^{c}}{m}\right]^2 \right),$$ где

$$\sum_{in}^{c} = \sum_{u,v \in c,\ e(u,v) \in E} w(u,v),$$

$\sum_{in}^{c}$ - Сумма весов всех внутренних рёбер сообщества

$$\sum_{tot}^{c} = \sum_{in}^{c} + \sum_{u \in c, v \notin c, e(u,v) \in E} w(u,v),$$

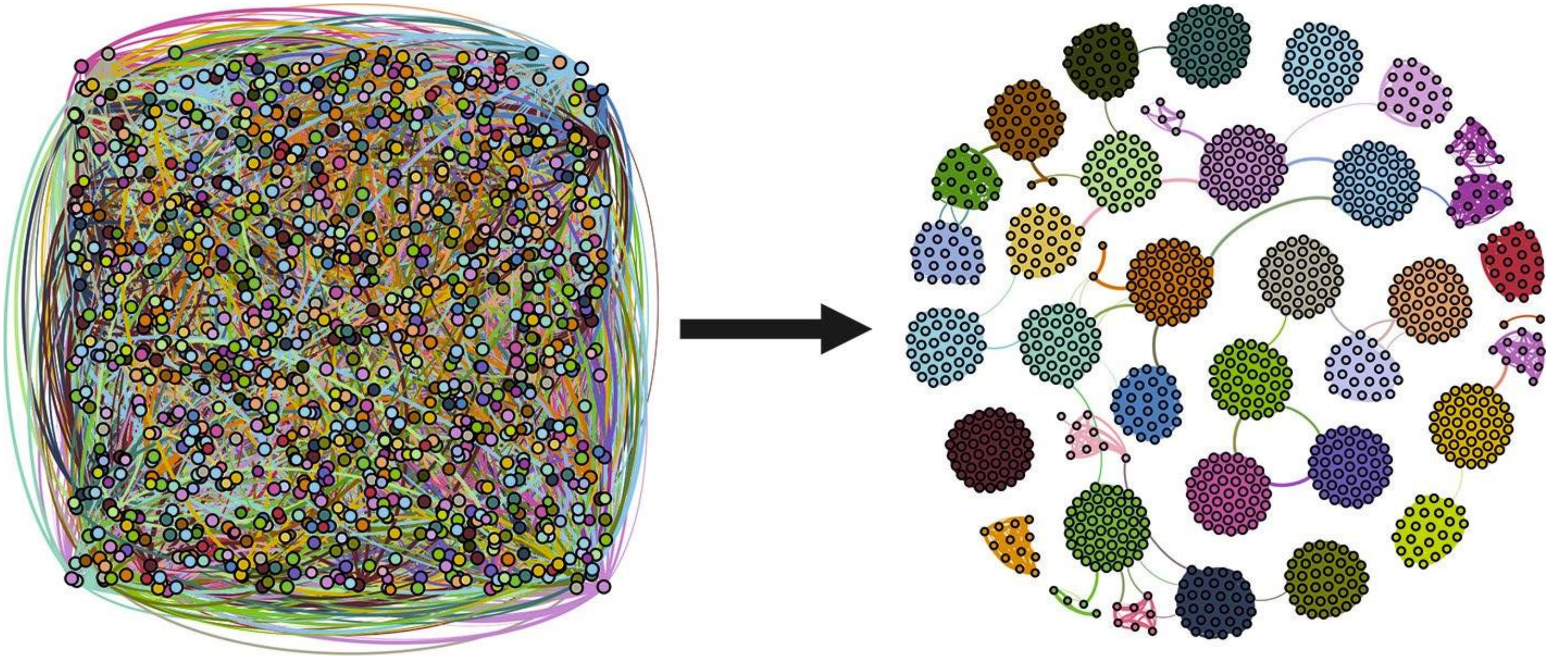
$\sum_{tot}^{c}$ - Сумма весов всех рёбер, исходящих из вершин сообщества



$$m = \sum_{e(u,v) \in E} w(u,v).$$

m – Нормализующий коэффициент, сумма весов всех рёбер графа

Community Detection



SSCA2, N=2^{10}, http://gephi.org

Обзор алгоритмов решения задачи Community Detection

Задача оптимизации модулярности – NP-полная

> 200 алгоритмов

- **Лувенский (Louvain)** $O(m)$
- **Label Propagion** $O(m)$
- **Girvan Newman** $O(nm^2)$
- **Clauset** $O(m\log^2 n)$
- **Рандомизированные алгоритмы**
- **Многоуровневые алгоритмы**

Лувенский алгоритм

- Blondel et al., University of Louvain, 2008
- Локальный жадный алгоритм
- Bottom-up multilevel (восходящий многоуровневый)
- O(m)

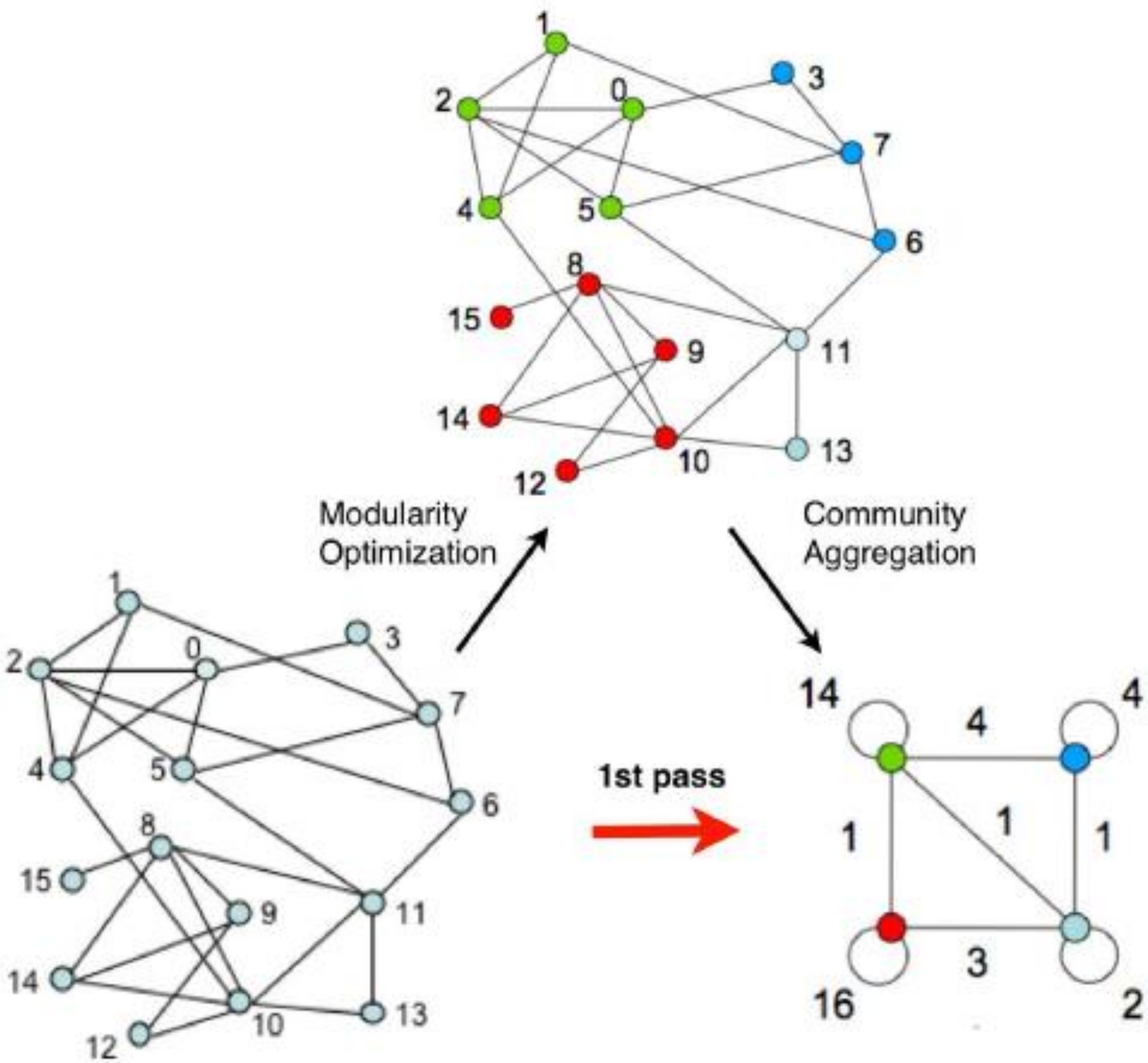
Лувенский алгоритм

```
do
    changed = false
    // Фаза 1. Перемещения
    // Фаза 2. Сжатие
        G(V, E) = compress(G)
while changed
```

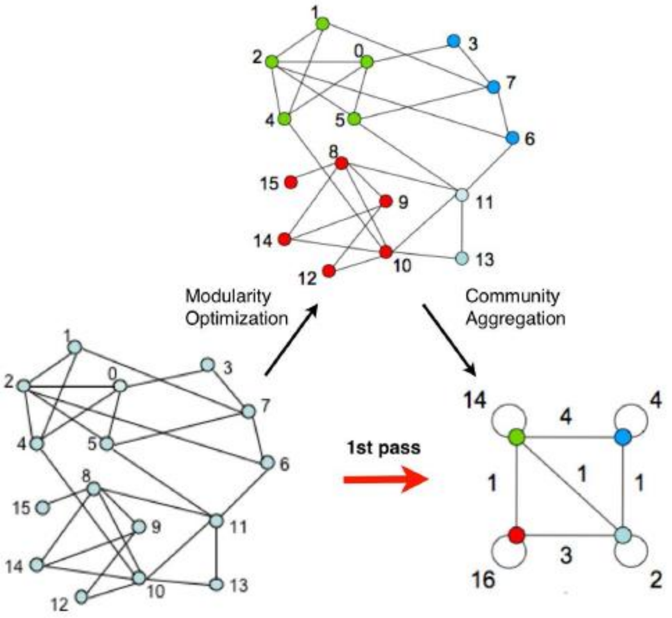
Лувенский алгоритм (2)

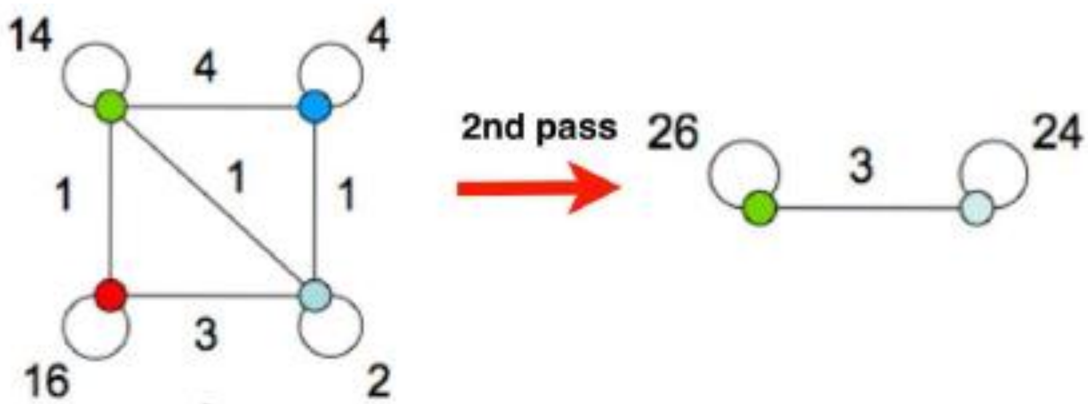
```
c[u]=u, u∈V
// Фаза 1. Перемещения
do
    more = false
    for all u ∈V
        best_component = argmax_c
                [Δmod(u->c : ∀c[v], (u,v)∈E, c[v] ≠ c[u], Δmod > 0)]
        if c[u] ≠ best_component
            update()
            c[u] = best_component
            more = true
        end if
    end for
    changed = changed | more
while more
```

Лувенский алгоритм (3)



Лувенский алгоритм (4)





Алгоритм Label Propagation

- Raghavan, USA, 2007
- O(m)
- Количество итераций слабо зависит от размера графа
- Очень прост
- Нет уникального решения
- Иногда может давать плохое решение – одно большое сообщество

Алгоритм Label Propagation

component_id[u]=u, u∈V

do

updated = false

for all u ∈V in **random order**

$l = \text{argmax}_c\ [\Sigma_c\ w(e), \forall c, e(u,v) \in E, v \in c]$

if component_id[u] ≠ l

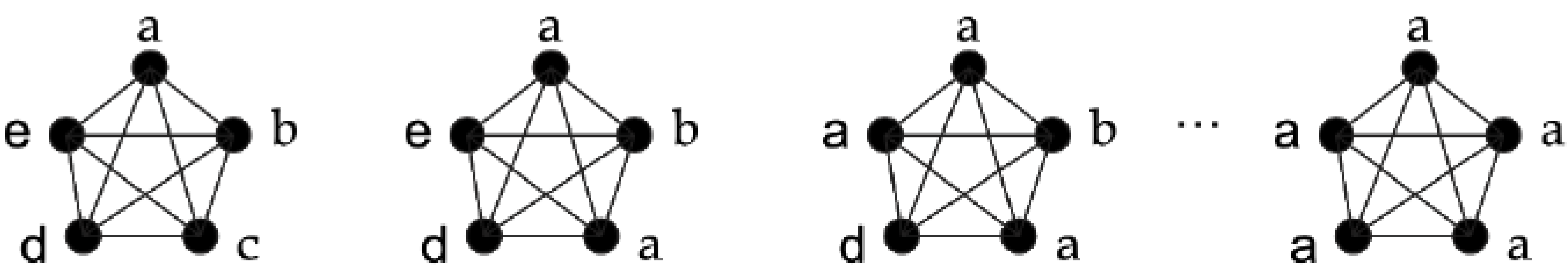
component_id[u] = l

updated = true

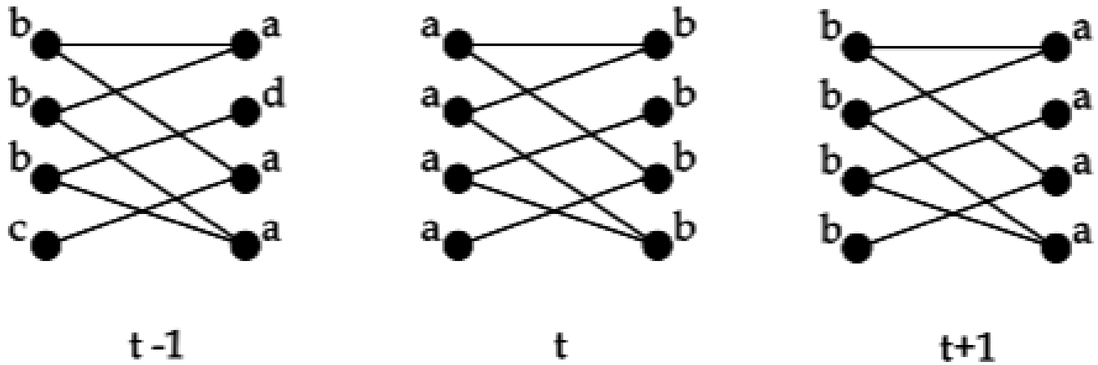
end if

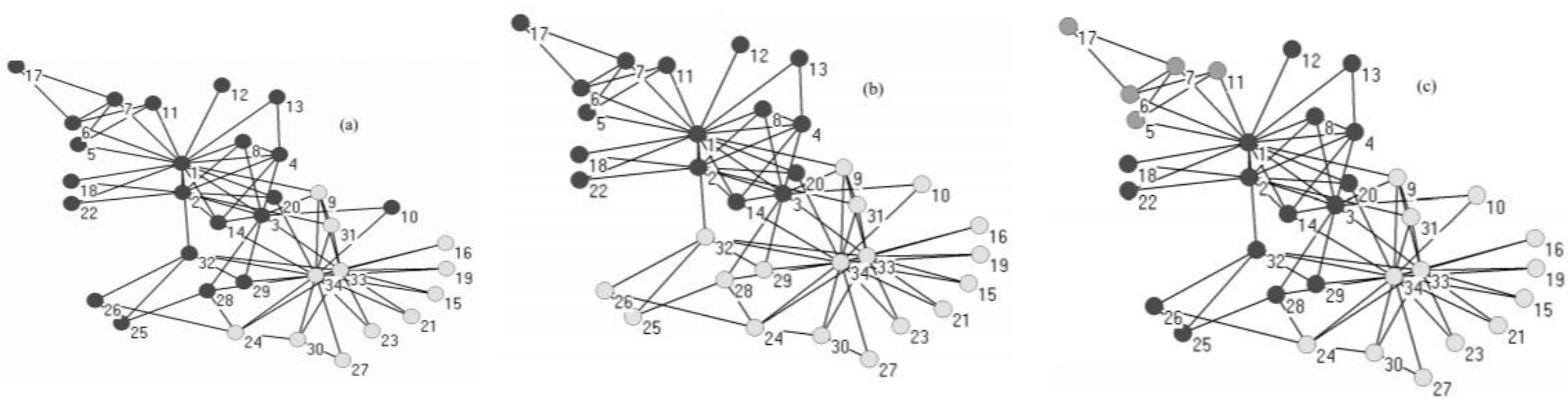
end for

while updated



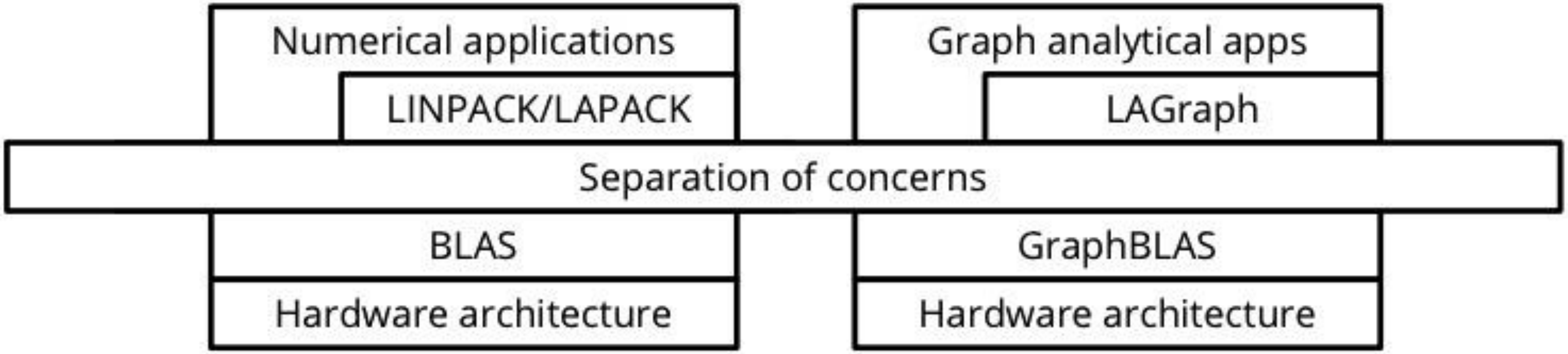
Алгоритм Label Propagation





Graph BLAS

- Идея представлять графы матрицами [D. Konig, 1931, 1936]
- 1960-е: Матричная алгебра – полезный инструмент для анализа графов
- 1979: BLAS Basic Linear Algebra Supbrograms (плотные)
- Графовые алгоритмы и sparse linear algebra [Kepner & Gilbert 2011]
- 2013: GraphBLAS https://graphblas.github.io/



Скалярные операции GraphBLAS

- Операции $\oplus \otimes$
- Коммутативность $a.b = b.a$
- Ассоциативность $a.(b.c) = (a.b).a$
- Дистрибутивность $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$
- 0 элемент

$$a \oplus 0 = a$$

$$a \otimes 0 = 0$$

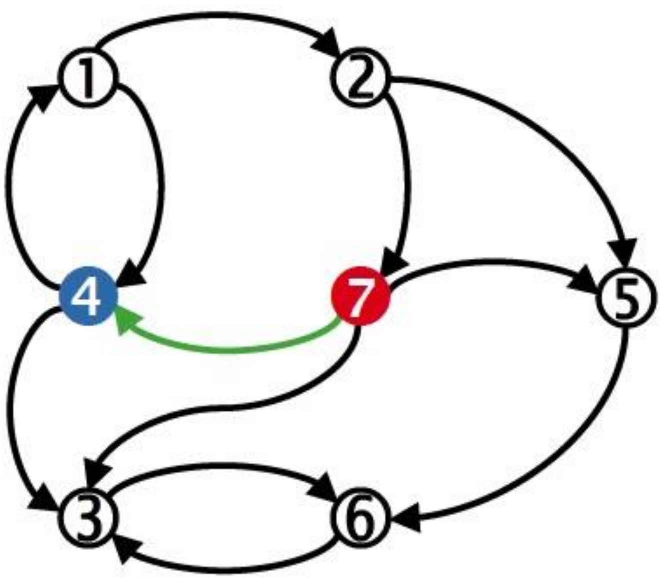
Примеры

semiring	domain	$\oplus$	$\otimes$	0
integer arithmetic	$a \in \mathbb{N}$	$+$	$\cdot$	0
real arithmetic	$a \in \mathbb{R}$	$+$	$\cdot$	0
lor-land	$a \in \{\mathrm{F}, \mathrm{T}\}$	$\vee$	$\wedge$	F
Galois field	$a \in \{0,1\}$	xor	$\wedge$	0
power set	$a \subset \mathbb{Z}$	$\cup$	$\cap$	$\emptyset$

https://archive.fosdem.org/2020/schedule/event/graphblas/attachments/slides/4053/export/events/attachments/graphblas/slides/4053/graphblas_fosdem_2020.pdf

Разреженная матрица

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{if } (v_i, v_j) \notin E \end{cases}$$



Most cells are zero:
sparse matrix

target

A	①	②	③	④	⑤	⑥	⑦
①		1		1			
②					1		1
③						1	
④	1		1				
⑤						1	
⑥			1				
source ⑦			1	1	1		

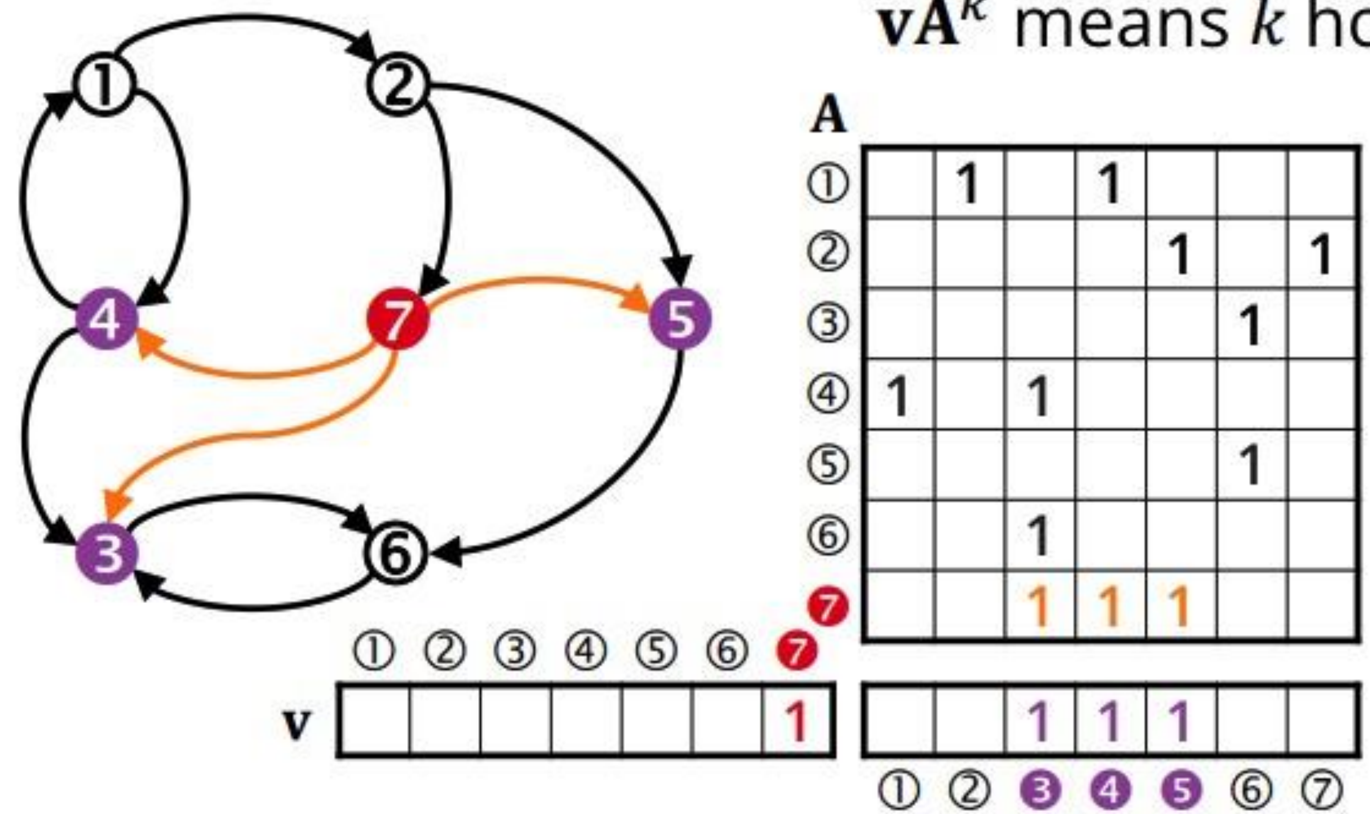
Базовые операции Graph BLAS

Функция	Аргументы	Результат	Семантика
MxV (SpMV)	- разреж. матрица A - разреж.(плотный) вектор x	- разреж (плотный) вектор y	y = A * x
MxM (SpGEMM)	- разреж. матрицы A, B	- разреж. матрица C	C = A * B
BuildMatrix	- списки (i, j, v)	- разреж. матрица A	Граф A = sparse(i, j, v)
GetTuples	- разреж. матрица A	- списки (i, j, v)	[i, j, v] = GetTuples(A)
Extract	- разреж. матрица A - индекс-векторы p, q	- разреж. матрица B	Подграф B = A(p, q)
Transpose	- разреж. матрица A	- разреж. матрица B	$B = A^T$
Modify	- разреж. матрицы A, B - индекс-векторы p, q	–	A(p,q) = B
EWiseMult, Add	- разреж. матрицы (вектора) A, B	- разреж. матрица (вектор) C	C = A .* B C = A + B

SpMV

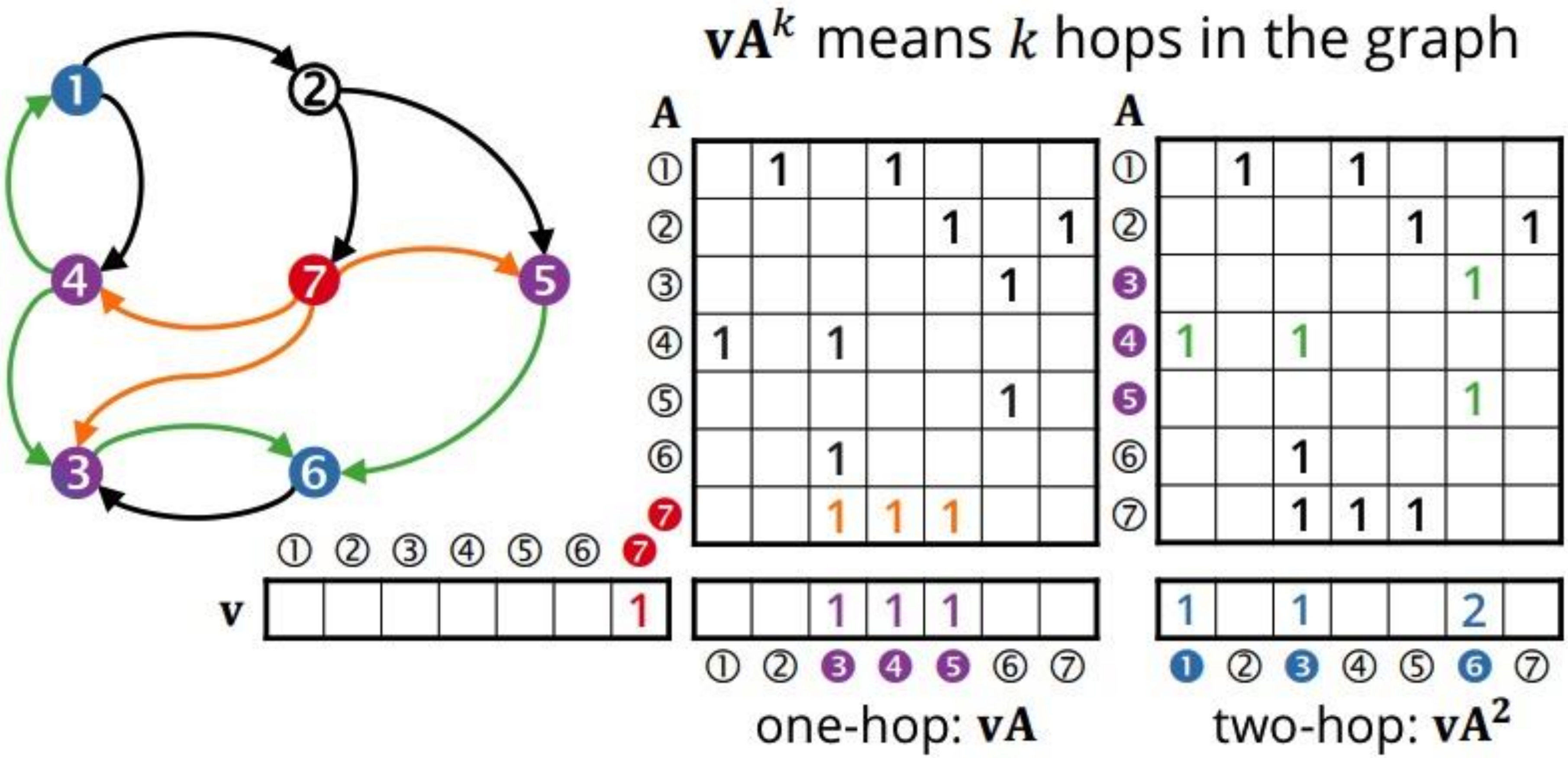
semiring	domain	$\oplus$	$\otimes$	0
integer arithmetic	$a \in \mathbb{N}$	$+$	$\cdot$	0

$\mathbf{vA}^k$ means k hops in the graph



one-hop: $\mathbf{vA}$

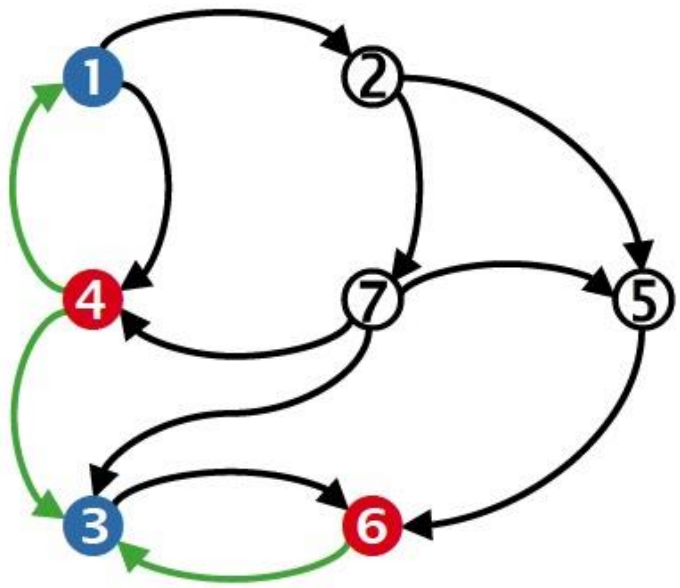
SpMV

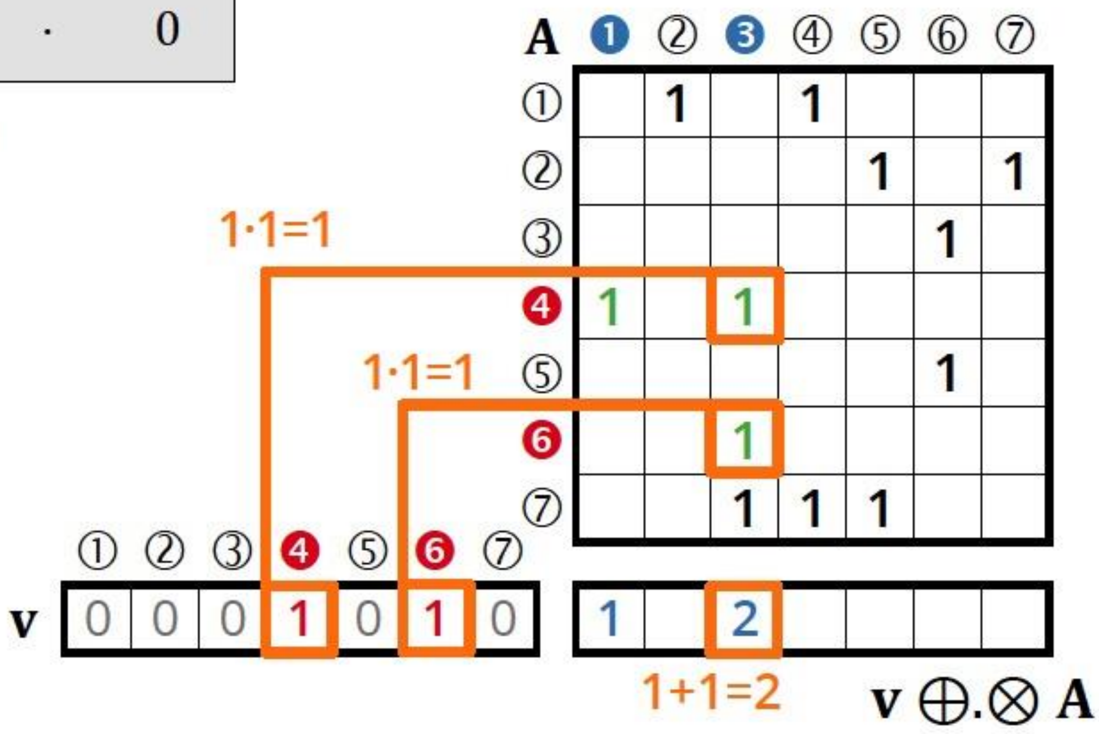


SpMV, integer

semiring	domain	⊕	⊗	0
integer arithmetic	$a \in \mathbb{N}$	$+$	$\cdot$	0

Semantics: **number of paths**

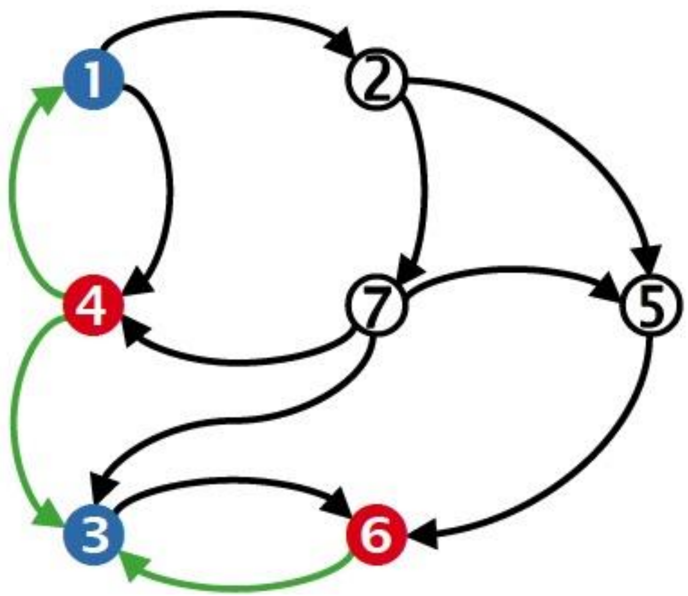


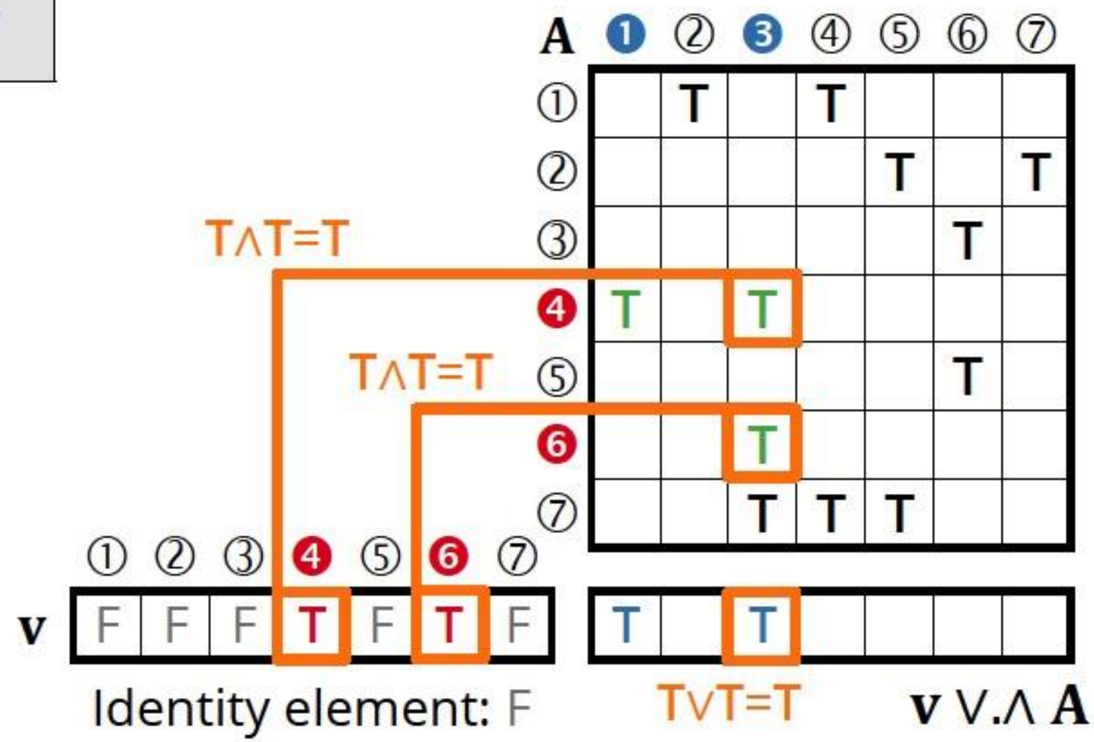


SpMV, boolean

semiring	domain	⊕	⊗	0
lor-land	$a \in \{F, T\}$	$\vee$	$\wedge$	F

Semantics: **reachability**

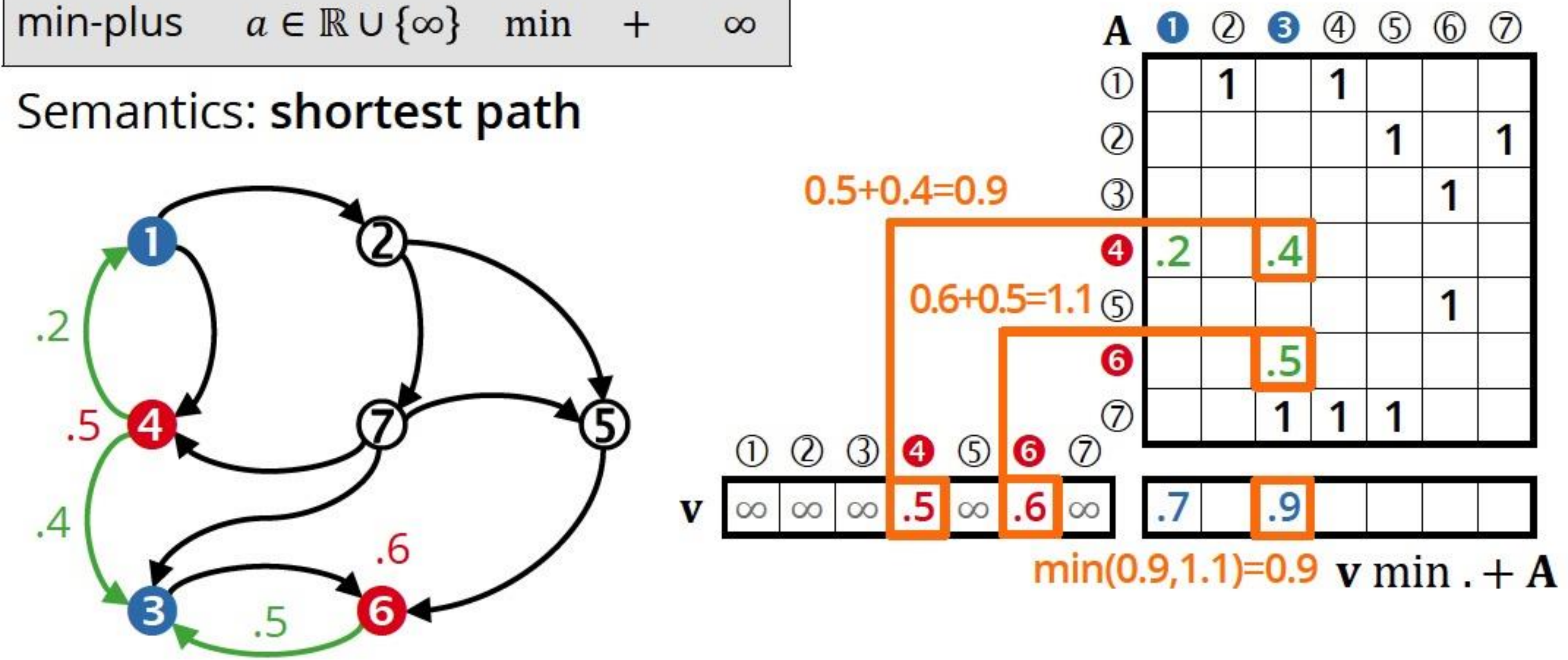




SpMV, min-plus

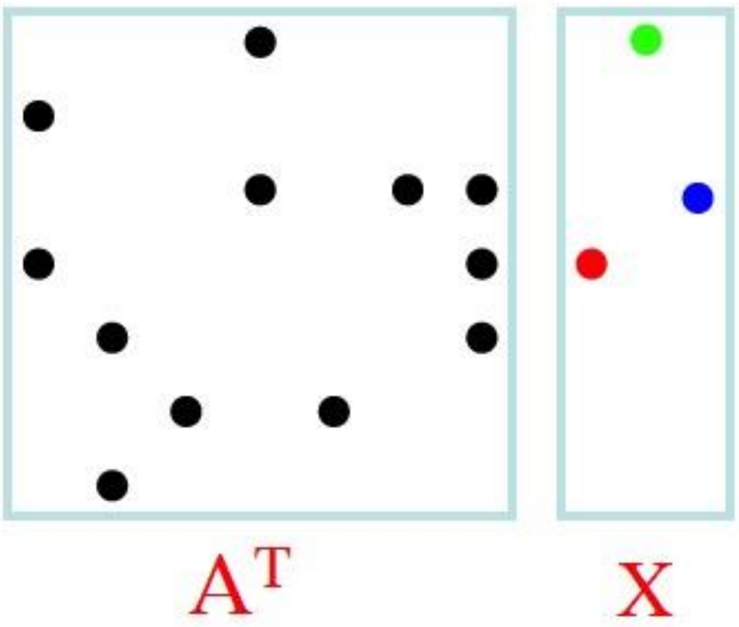
semiring	domain	⊕	⊗	0
min-plus	$a \in \mathbb{R} \cup \{\infty\}$	min	+	∞

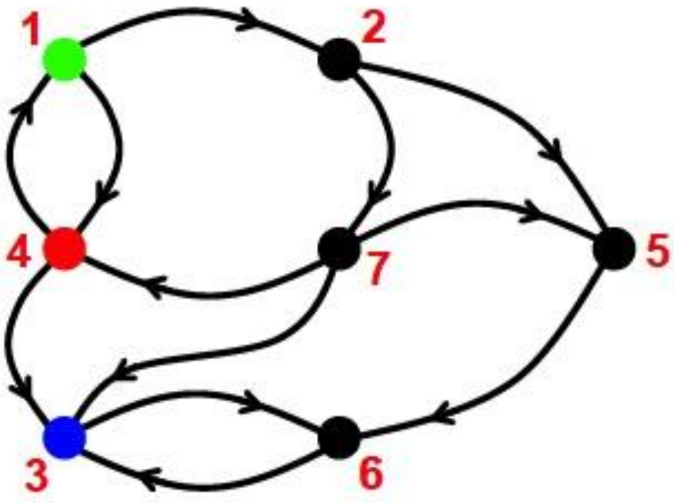
Semantics: **shortest path**



SpGEMM

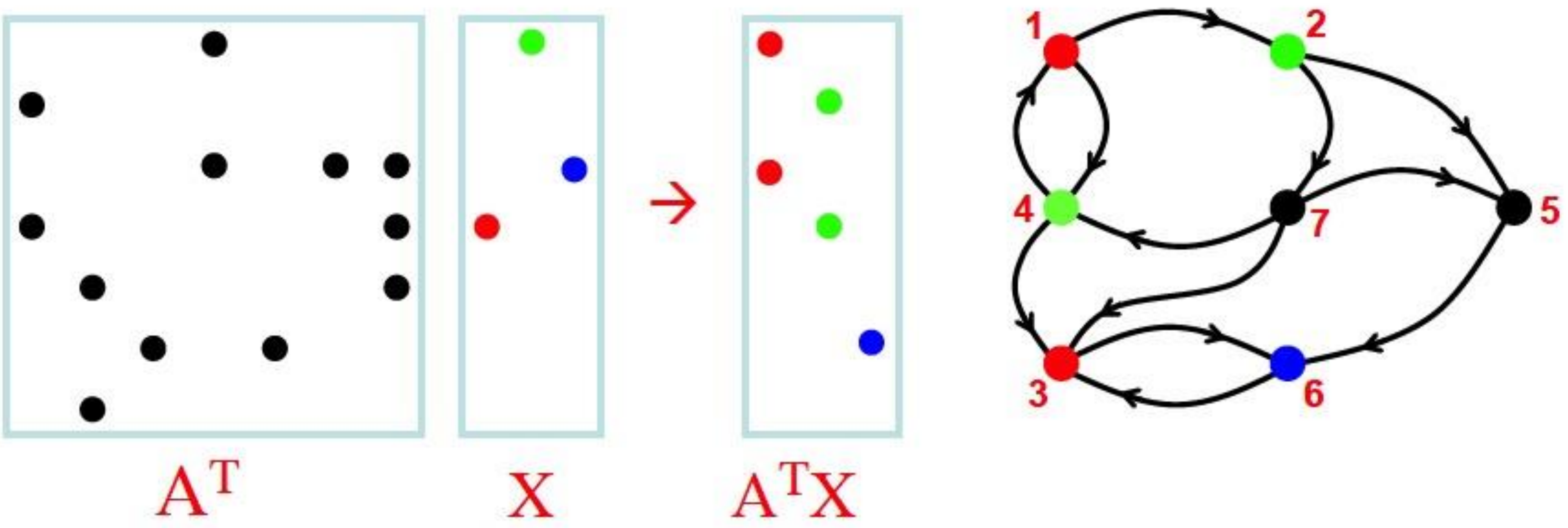
BFS от нескольких источников





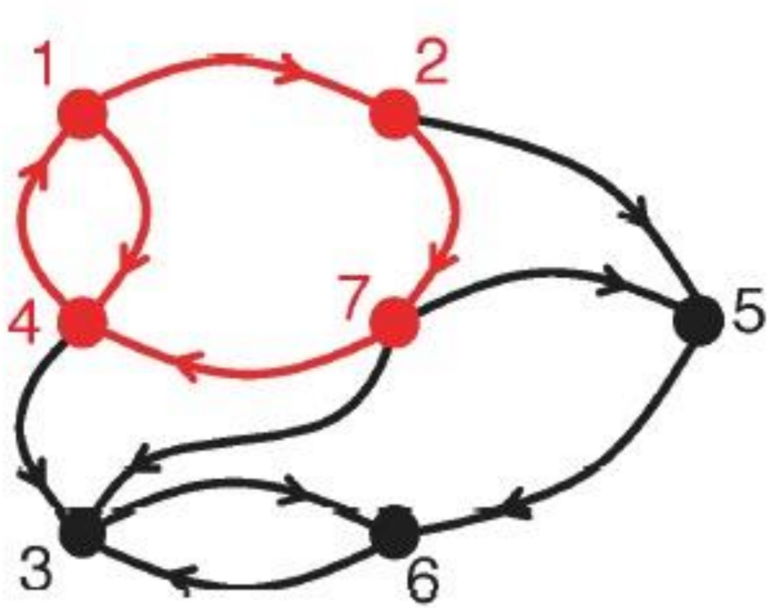
SpGEMM

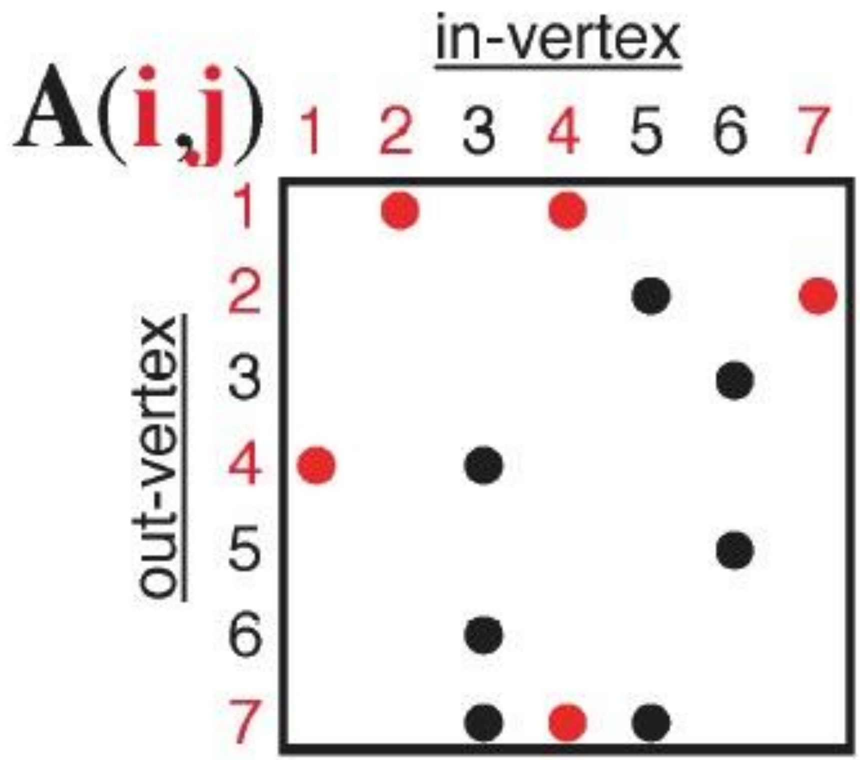
BFS от нескольких источников



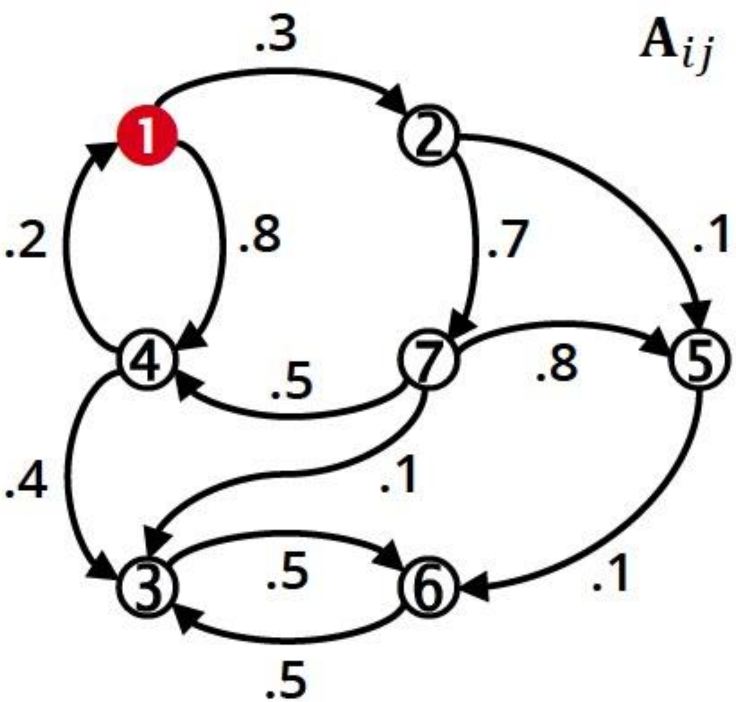
Extract

i = j = {1, 2, 4, 7}





SSSP, алгоритм Беллмана-Форда



$$\mathbf{A}_{ij} = \begin{cases} 0 & \text{if } i = j \\ w(e_{ij}) & \text{if } e_{ij} \in E \\ \infty & \text{if } e_{ij} \notin E \end{cases}$$

$$\mathbf{d} = [\infty\ \infty \dots \infty]$$

$$\mathbf{d}(s) = 0$$

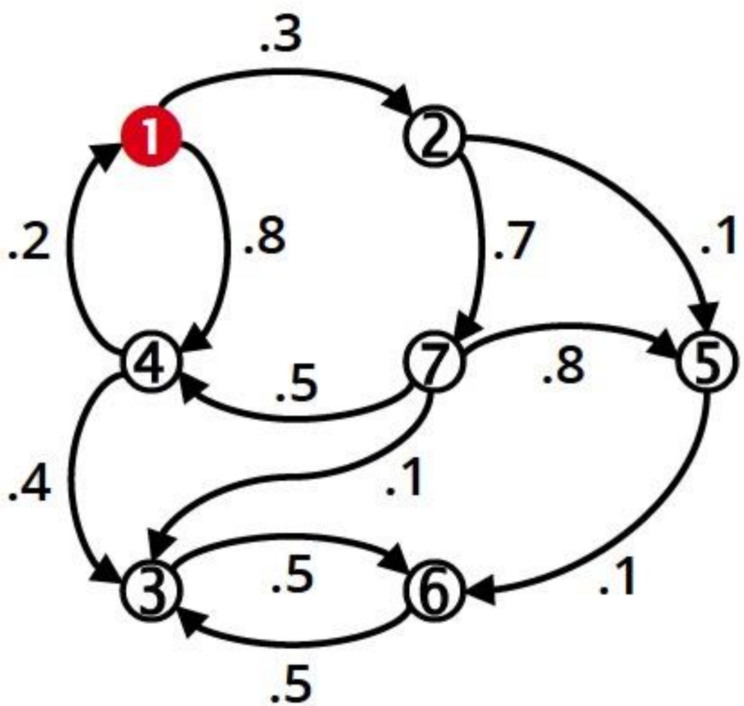
A	①	②	③	④	⑤	⑥	⑦
①	0	.3	∞	.8	∞	∞	∞
②	∞	0	∞	∞	.1	∞	.7
③	∞	∞	0	∞	∞	.5	∞
④	.2	∞	.4	0	∞	∞	∞
⑤	∞	∞	∞	∞	0	.1	∞
⑥	∞	∞	.5	∞	∞	0	∞
⑦	∞	∞	.1	.5	.9	∞	0

	❶	②	③	④	⑤	⑥	⑦
d	0	∞	∞	∞	∞	∞	∞

d min.+ **A**

SSSP, алгоритм Беллмана-Форда

semiring	set	$\oplus$	$\otimes$	0
min-plus	$a \in \mathbb{R} \cup \{\infty\}$	min	+	∞



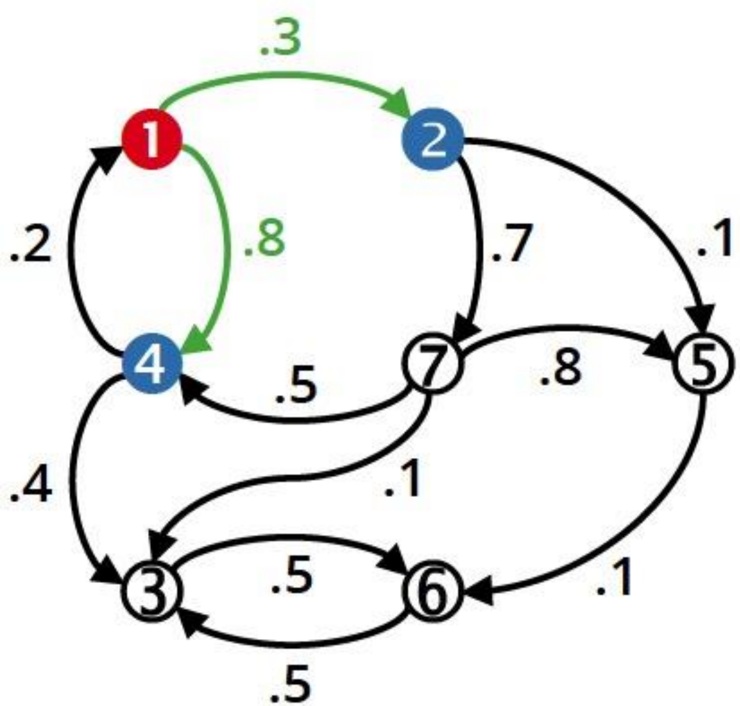
A	①	②	③	④	⑤	⑥	⑦
①	0	.3		.8			
②		0			.1		.7
③			0			.5	
④	.2		.4	0			
⑤					0	.1	
⑥			.5			0	
⑦			.1	.5	.9		0

	1	②	③	④	⑤	⑥	⑦
d	0	∞	∞	∞	∞	∞	∞

d min.+ **A**

SSSP, алгоритм Беллмана-Форда

semiring	set	⊕	⊗	0
min-plus	$a \in \mathbb{R} \cup \{\infty\}$	min	+	∞



A	①	②	③	④	⑤	⑥	⑦
①	0	.3		.8			
②		0			.1		.7
③			0			.5	
④	.2		.4	0			
⑤					0	.1	
⑥			.5			0	
⑦			.1	.5	.9		0

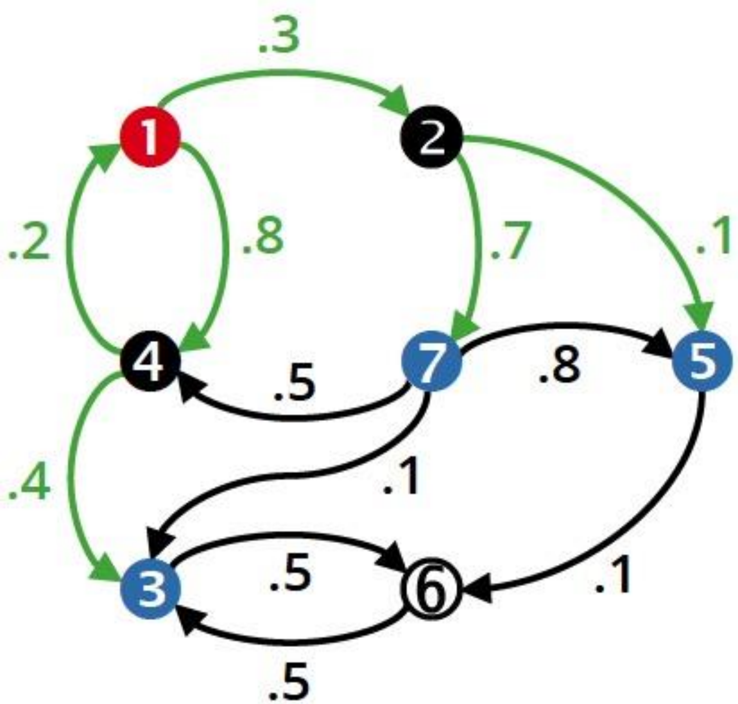
	❶	②	③	④	⑤	⑥	⑦
d	0	∞	∞	∞	∞	∞	∞

0	.3		.8			

d min.+ **A**

SSSP, алгоритм Беллмана-Форда

semiring	set	$\oplus$	$\otimes$	0
min-plus	$a \in \mathbb{R} \cup \{\infty\}$	min	+	∞

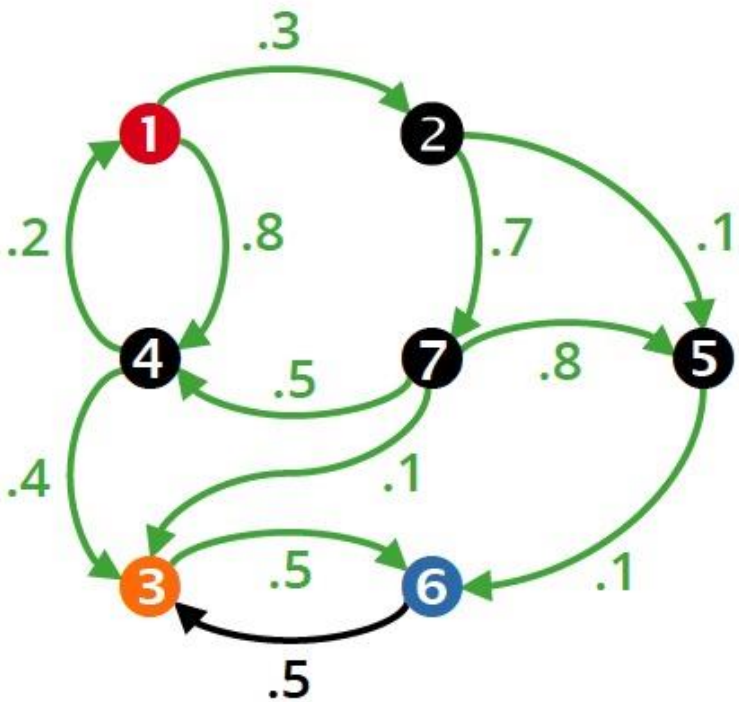


A	①	②	③	④	⑤	⑥	⑦
①	0	.3		.8			
②		0			.1		.7
③			0			.5	
④	.2		.4	0			
⑤					0	.1	
⑥			.5			0	
⑦			.1	.5	.9		0

	❶	❷	③	❹	⑤	⑥	⑦
d	0	.3	∞	.8	∞	∞	∞
d min.+ **A**	0	.3	1.2	.8	.4		1

SSSP, алгоритм Беллмана-Форда

semiring	set	⊕	⊗	0
min-plus	$a \in \mathbb{R} \cup \{\infty\}$	min	+	∞

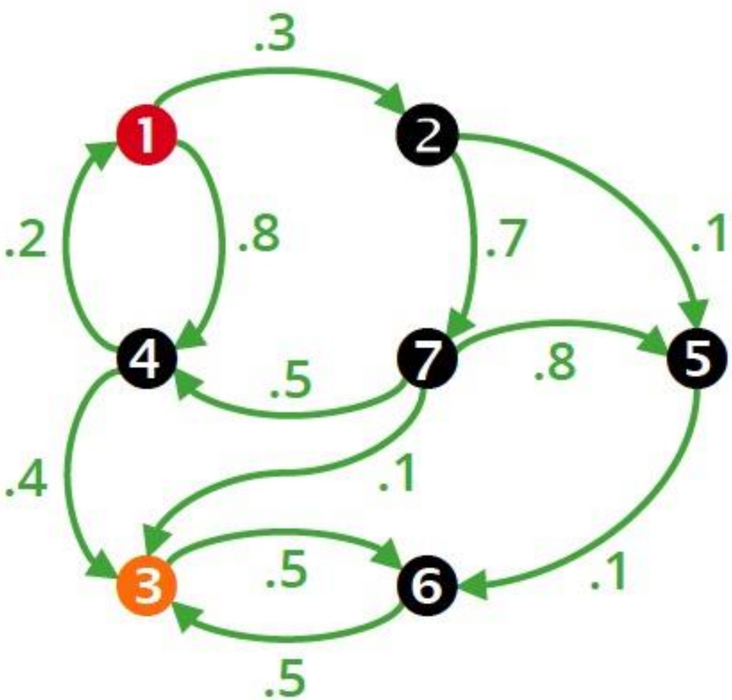


A	①	②	③	④	⑤	⑥	⑦
①	0	.3		.8			
②		0			.1		.7
③			0			.5	
④	.2		.4	0			
⑤					0	.1	
⑥			.5			0	
⑦			.1	.5	.9		0

	1	2	3	4	5	⑥	7
d	0	.3	1.2	.8	.4	∞	1
d min.+ A	0	.3	1.1	.8	.4	.5	1

SSSP, алгоритм Беллмана-Форда

semiring	set	$\oplus$	$\otimes$	0
min-plus	$a \in \mathbb{R} \cup \{\infty\}$	min	$+$	∞



A	①	②	③	④	⑤	⑥	⑦
①	0	.3		.8			
②		0			.1		.7
③			0			.5	
④	.2		.4	0			
⑤					0	.1	
⑥			.5			0	
⑦			.1	.5	.9		0

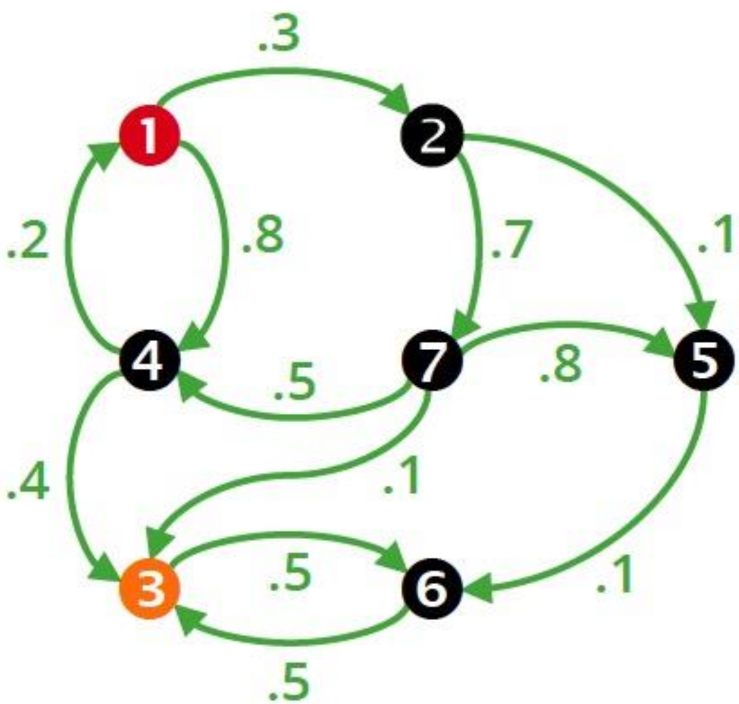
	1	2	3	4	5	6	7
d	0	.3	1.1	.8	.4	.5	1

0	.3	1	.8	.4	.5	1

d min.+ **A**

SSSP, алгоритм Беллмана-Форда

semiring	set	$\oplus$	$\otimes$	0
min-plus	$a \in \mathbb{R} \cup \{\infty\}$	min	+	∞



A	①	②	③	④	⑤	⑥	⑦
①	0	.3		.8			
②		0			.1		.7
③			0			.5	
④	.2		.4	0			
⑤					0	.1	
⑥			.5			0	
⑦			.1	.5	.9		0

	1	2	3	4	5	6	7
d	0	.3	1	.8	.4	.5	1

0	.3	1	.8	.4	.5	1

d min.+ **A**

SSSP, алгоритм Беллмана-Форда

Алгоритм:

```
d = [inf, inf, ..., inf]
d(s) = 0
for k = 1 to N-1
    d = d min.+ A
```

Graph BLAS

Стандарт

- The GraphBLAS C API Specification v.2.0, 2021/11/15
  https://graphblas.org/docs/GraphBLAS_API_C_v2.0.0.pdf

Реализации

- SuiteSparse:GraphBLAS v. 9.0.3. 2024/03/01
  https://github.com/DrTimothyAldenDavis/GraphBLAS
- IBM Graph BLAS
- GraphBLAS Template Library (PNNL, Indiana University)
  - C++, C API interface
- GraphBLAST, для GPU

https://github.com/gunrock/graphblast

- Combinatorial BLAS, v. 1.6.2 MPI-реализация, UCSB
  https://github.com/PASSIONLab/CombBLAS