

BFS с использованием GraphBLAS

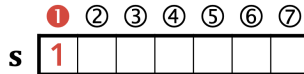
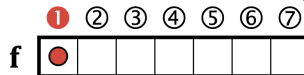
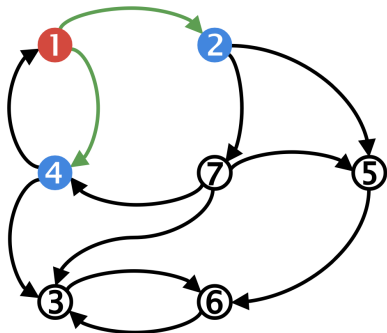
Рустам Азимов

- ▶ **BFS** — алгоритм обхода графа в ширину
- ▶ Хотя BFS является одним из простейших алгоритмов обхода графа, он служит основой для многих других важных алгоритмов анализа графов
- ▶ Формулировки
 - ▶ Levels — вычислить уровень каждой вершины, относительно стартовой
 - ▶ Parents — для каждой вершины v вычислить предыдущую вершину p на пути от стартовой вершины до вершины v (эта информация используется для восстановления путей)
 - ▶ MSBFS (Multiple-Source BFS) — начать обход из нескольких стартовых вершин

BFS-LEVELS

semiring	domain	\oplus	\otimes	0
any-pair	{T, F}	any	pair	F

level = 1



A

	1	2	3	4	5	6	7
1		●		●			
2					●		●
3						●	
4	●		●				
5						●	
6			●				
7			●	●	●		

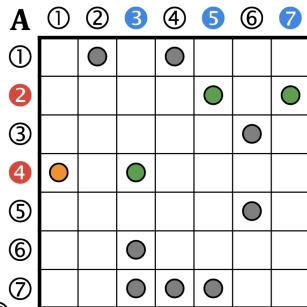
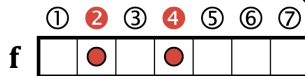
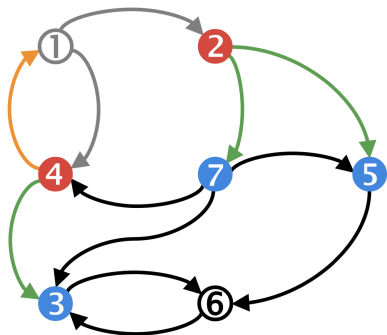


$f\langle \neg s \rangle = f$ any. pair **A**

BFS-LEVELS

semiring	domain	\oplus	\otimes	0
any-pair	{T, F}	any	pair	F

level = 2

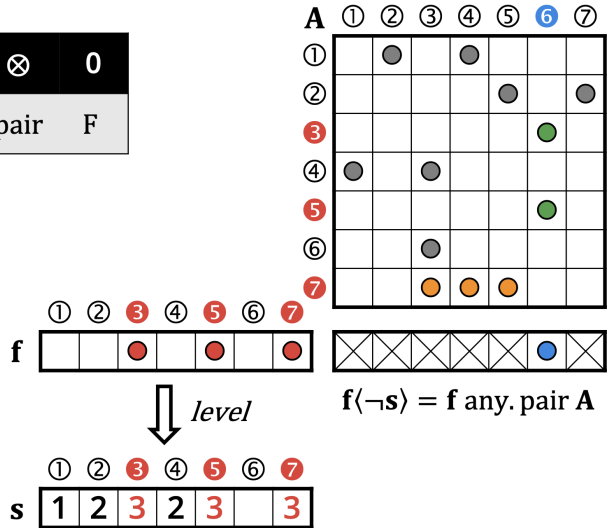
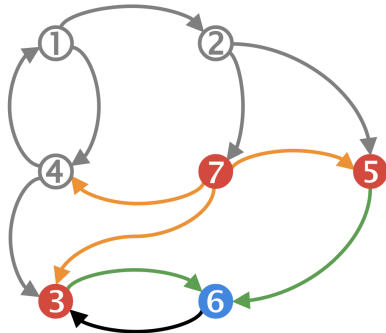


$f\langle \neg s \rangle = f$ any. pair A

BFS-LEVELS

semiring	domain	\oplus	\otimes	0
any-pair	{T, F}	any	pair	F

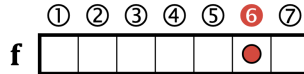
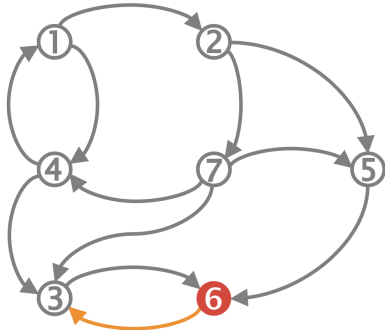
level = 3



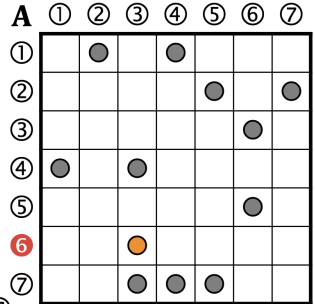
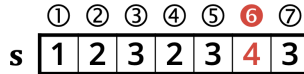
BFS-LEVELS

semiring	domain	\oplus	\otimes	0
any-pair	{T, F}	any	pair	F

level = 4



↓ *level*



$f \langle \neg s \rangle = f$ any. pair **A**

f is empty
→ terminate

BFS-LEVELS: ALGORITHM

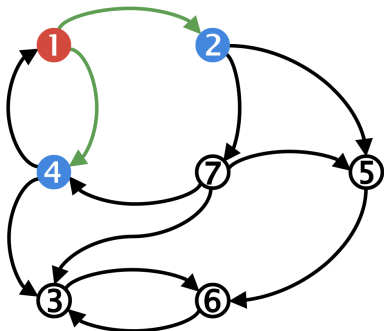
- **Input:** adjacency matrix \mathbf{A} , source vertex s , #vertices n
- **Output:** vector of visited vertices \mathbf{v} (integer)
- **Workspace:** frontier vector \mathbf{f} (Boolean)

1. $\mathbf{f}(s) = \text{T}$
2. for $level = 1$ to $n - 1$ *terminate earlier if \mathbf{f} is empty
3. $\mathbf{s}(\mathbf{f}) = level$ assign the level value to the vertices in the frontier
4. $\text{clear}(\mathbf{f})$ clear the frontier \mathbf{f}
5. $\mathbf{f}(\neg \mathbf{s}) = \mathbf{f}$ any. pair \mathbf{A} advance the frontier

BFS-PARENTS

semiring	domain	\oplus	\otimes	0
min-first	\mathbb{N}	min	first	0

$$\text{first}(x, y) = x$$



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

f	①	②	③	④	⑤	⑥	⑦
	1						

⊗	1		1				
---	---	--	---	--	--	--	--

$$f \langle \neg p \rangle = f \min . \text{first } A$$

$$\Downarrow p \langle f \rangle = f$$

$$\Downarrow f \langle f \rangle = \text{id}$$

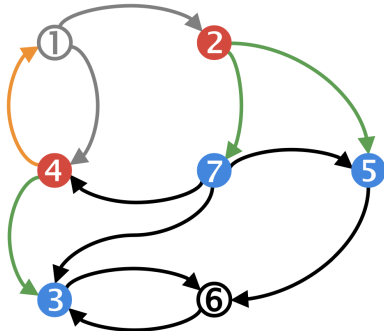
p	①	②	③	④	⑤	⑥	⑦
	0	1		1			

	①	②	③	④	⑤	⑥	⑦
		2		4			

BFS-PARENTS

semiring	domain	\oplus	\otimes	0
min-first	\mathbb{N}	min	first	0

$$\text{first}(x, y) = x$$



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

f	①	②	③	④	⑤	⑥	⑦
		2		4			

$f \langle \neg p \rangle$	×	×	4	×	2		2
----------------------------	---	---	---	---	---	--	---

$$f \langle \neg p \rangle = f \min . \text{first } A$$

$$\Downarrow p \langle f \rangle = f$$

$$\Downarrow f \langle f \rangle = \text{id}$$

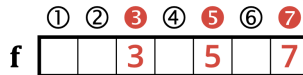
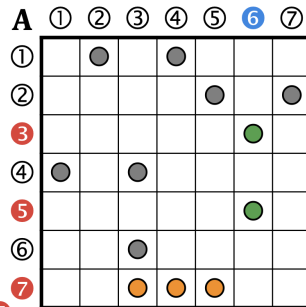
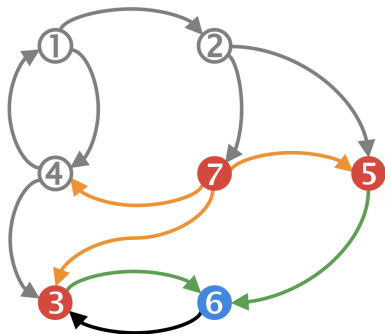
p	①	②	③	④	⑤	⑥	⑦
	0	1	4	1	2		2

	①	②	③	④	⑤	⑥	⑦
			3		5		7

BFS-PARENTS

semiring	domain	\oplus	\otimes	0
min-first	\mathbb{N}	min	first	0

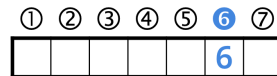
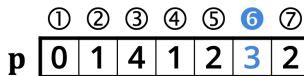
$$\text{first}(x, y) = x$$



$$f\langle \neg p \rangle = f \min . \text{first } A$$

$$\Downarrow p\langle f \rangle = f$$

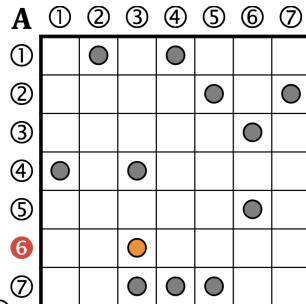
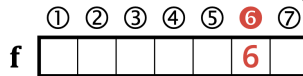
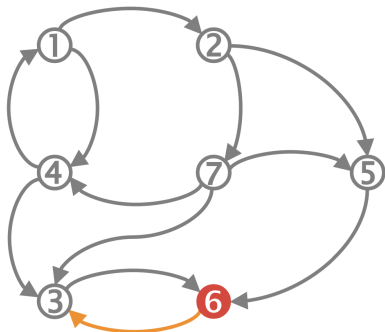
$$\Downarrow f\langle f \rangle = \text{id}$$



BFS-PARENTS

semiring	domain	\oplus	\otimes	0
min-first	\mathbb{N}	min	first	0

$$\text{first}(x, y) = x$$



$$f\langle \neg p \rangle = f \text{ min } . \text{first } A$$

f is empty
→ terminate

BFS-PARENTS: ALGORITHM

- **Input:** adjacency matrix \mathbf{A} , source vertex s , #vertices n
- **Output:** parent vertices vector \mathbf{p} (integer)
- **Workspace:** vertex index vector \mathbf{idx} (integer), frontier vector \mathbf{f} (integer)

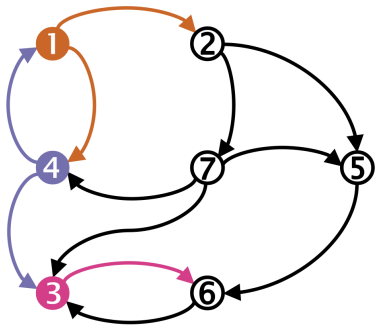
1. $\mathbf{idx} = [1 \ 2 \ \dots \ n]$ we assume 1-based indexing here
2. $\mathbf{f}(s) = s$
3. $\mathbf{p}(s) = 0$
4. for $l = 1$ to $n - 1$ *terminate earlier if the frontier is empty
5. $\mathbf{f}(\neg \mathbf{p}) = \mathbf{f} \text{ min. first } \mathbf{A}$ advance the frontier
6. $\mathbf{p}(\mathbf{f}) = \mathbf{f}$ assign parent ids to the frontier's vertices
7. $\mathbf{f}(\mathbf{f}) = \mathbf{idx}$ assign vertex ids $\mathbf{f}(i) = i$

- ▶ Использование операции **min** в качестве операции сложения позволяет выбрать из нескольких возможных предыдущих вершин и обеспечивает детерминированность
- ▶ Можно использовать другие операции, например полукольцо **any.first** для выбора произвольного **parent**

- ▶ Мы использовали вектор f в качестве текущего фронта
- ▶ Фронт инициализировался с одной стартовой вершиной
- ▶ Что будет, если стартовых вершин будет несколько?
- ▶ Какие будут проблемы?

MSBFS-LEVELS

semiring	domain	\oplus	\otimes	0
any-pair	{T, F}	any	pair	F



F	1	2	3	4	5	6	7
t1	●						
t2			●				
t3				●			

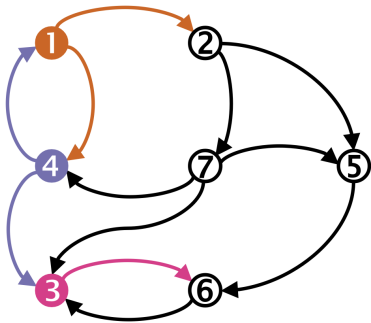
A	1	2	3	4	5	6	7
1		●		●			
2					●		●
3						●	
4	●		●				
5						●	
6			●				
7			●	●	●		

	●		●				
					●		
●		●					

$F\langle \neg S \rangle = F$ any. pair A

MSBFS-PARENTS

semiring	domain	\oplus	\otimes	0
min-first	$\mathbb{N} \cup \{+\infty\}$	min	first	$+\infty$



F	1	2	3	4	5	6	7
t1	1						
t2			3				
t3				4			

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

	1		1				
						3	
4		4					

$$F\langle \neg P \rangle = F \text{ min . first } A$$

DIRECTION-OPTIMIZING BFS

- ▶ Direction-optimizing BFS алгоритм был опубликован в 2012 году
 - ▶ S. Beamer, K. Asanovic, D. Patterson: Direction-Optimizing Breadth-First Search, SC 2012
- ▶ Направление обхода через матрично-векторное умножение меняется и иногда вычисляется, как vA (**push**), а иногда, как $A^T v$ (**pull**)
 - ▶ Использует **push** направление, когда вектор (фронт) сильно разрежен и имеет малое количество ненулевых значений
 - ▶ Использует **pull** направление, когда вектор (фронт) становится слишком плотным

- ▶ Книга: Кернер J., Гилберт J. — "Graph algorithms in the language of linear algebra"
- ▶ Презентация: Gábor Szárnyas — "Introduction to GraphBLAS"