



WYDZIAŁ
**MATEMATYKI
I FIZYKI STOSOWANEJ**
POLITECHNIKI RZESZOWSKIEJ

NAJKRÓTSZA DROGA

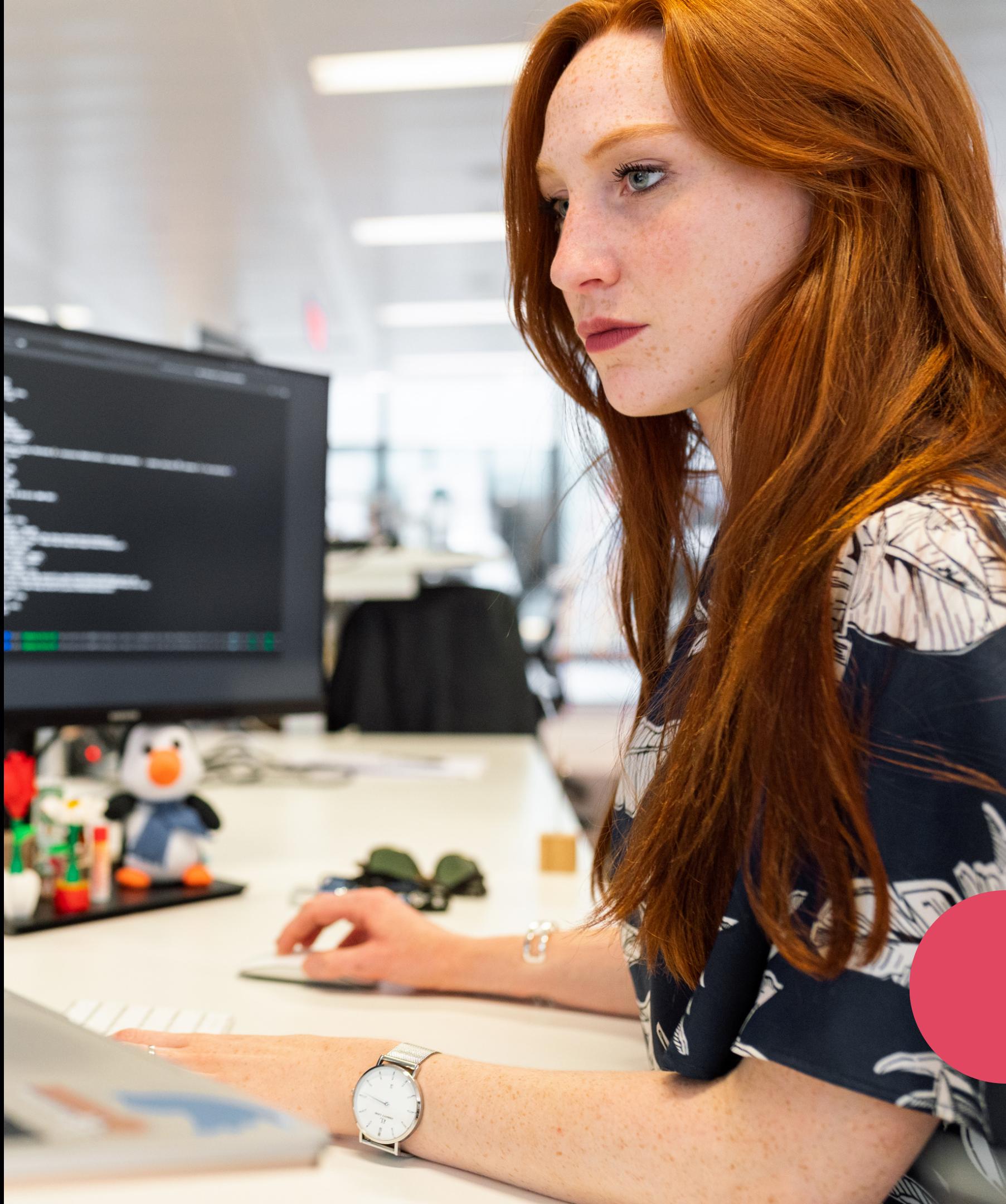
OPTYMALIZACJA DYSKRETNA
ZADANIE PROJEKTOWE

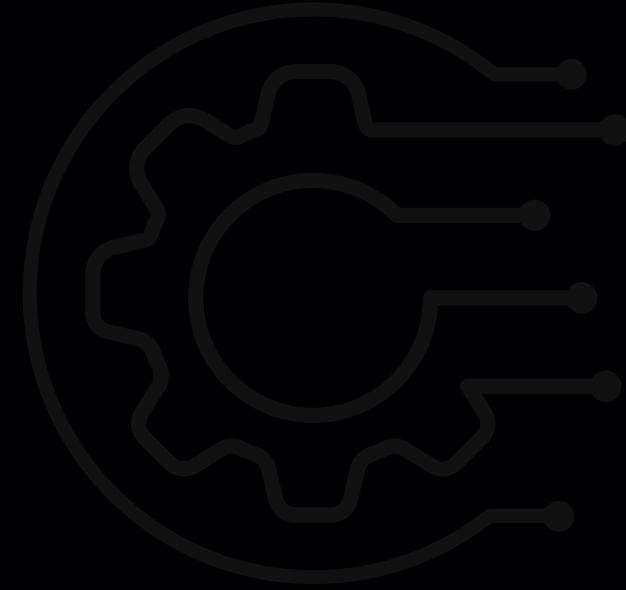
**Paweł Dziedzic
Jakub Frydlewicz**



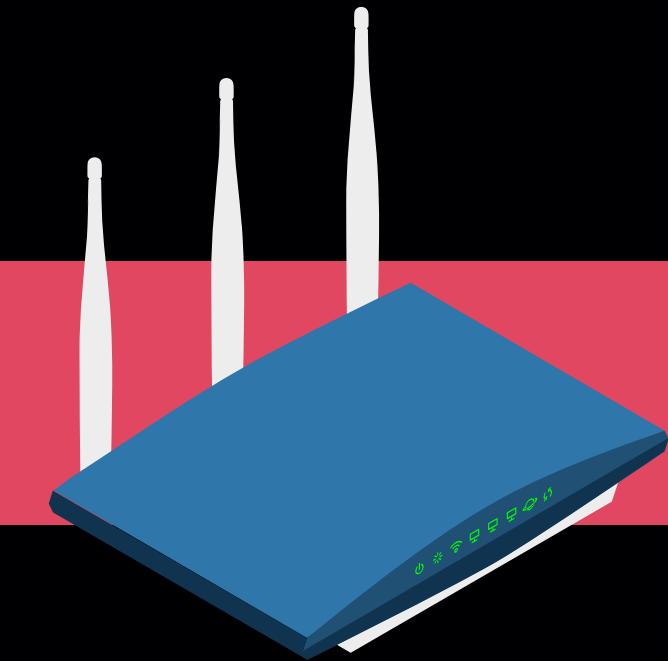
PROBLEM NAJKRÓTSZEJ ŚCIEŻKI

Najkrótsza droga to problem, który obecny jest w wielu dziedzinach, od logistyki po planowanie tras w nawigacji. Problem najkrótszej ścieżki polega na znalezieniu najkrótszej ścieżki pomiędzy dwoma wierzchołkami (lub węzłami) w grafie. Algorytmy takie jak algorytm Floyda-Warshalla i różne odmiany algorytmu Dijkstry są używane do znalezienia rozwiązań problemu najkrótszej ścieżki. Zastosowania problemu najkrótszej ścieżki obejmują te w sieciach drogowych, logistyce, komunikacji, projektowaniu elektronicznym, analizie awaryjnej sieci energetycznej i wykrywaniu społeczności.





WYKORZYSTANIE



algorytm Dijkstry

- sugerowanie listy znajomych na social mediach
- planowanie lotów dla klientów
- trasy dla robotów/dronów kurierów
- Wyznaczenie serwera plików w sieci LAN

algorytm Bellmana-Forda

- sieci telekomunikacyjne, trasowanie pakietów między węzłami sieci
- planowanie tras dla pojazdów
- modelowanie ryzyka i zarządzania portfelem
- renderowanie oświetlenia i cieni w grafice komputerowej

algorytm Floyda-Warshalla

- protokoły routingu (OSPF, IS-IS)
- system zarządzania flotą pojazdów (przypisywanie pojazdów do tras)
- zarządzanie projektami, minimalny czas wykonania projektu



ZALETY I WADY



algorytm Dijkstry

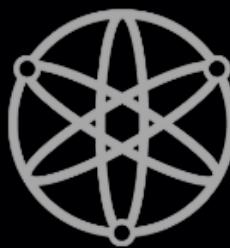
- + Złożoność prawie liniowa
- + Działa na grafach skierowanych, ważonych, wszystkie krawędzie powinny być nieujemne
- pochłania dużo czasu podczas przetwarzania.

algorytm Bellmana-Forda

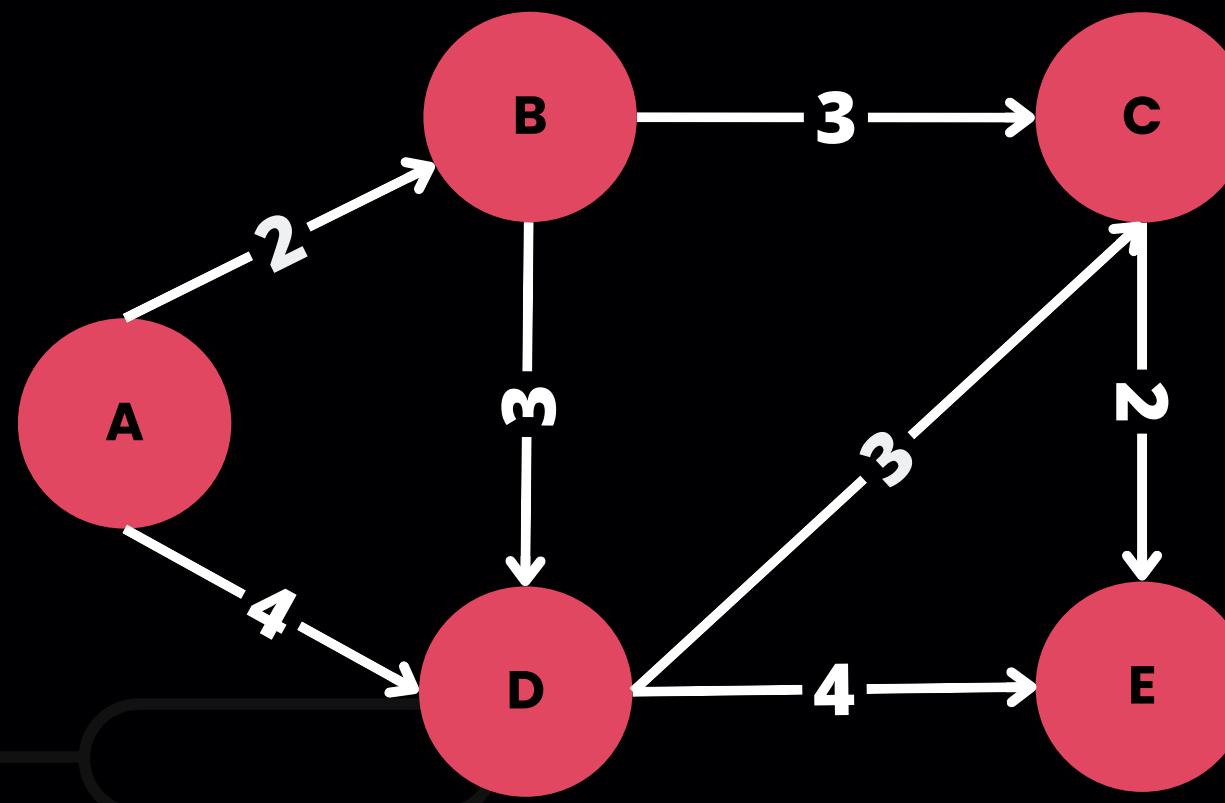
- + Algorytm Bellmana-Forda może obsługiwać krawędzie o ujemnych wagach.
- + Algorytm Bellmana-Forda może obsługiwać krawędzie o ujemnych wagach.
- Mniej wydajny niż algorytm Dijkstry dla grafów bez ujemnych wag.
- jeśli wykonuje zbyt wiele iteracji, może wpadać w pętlę i nie zakończyć działania.

algorytm Floyda-Warshalla

- + Może obsługiwać zarówno grafy z ujemnymi wagami, jak i znajdowanie najkrótszych ścieżek między dowolnymi parami wierzchołków.
- + Dla małych grafów algorytm Floyda-Warshalla może być efektywny i prosty do implementacji
- Mało wydajny dla większych grafów
- Wymaga większej ilości pamięci, ponieważ musi przechowywać macierz odległości dla wszystkich par wierzchołków.

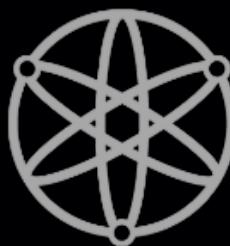


Najkrótsza ścieżka z jednego węzła do wszystkich węzłów.

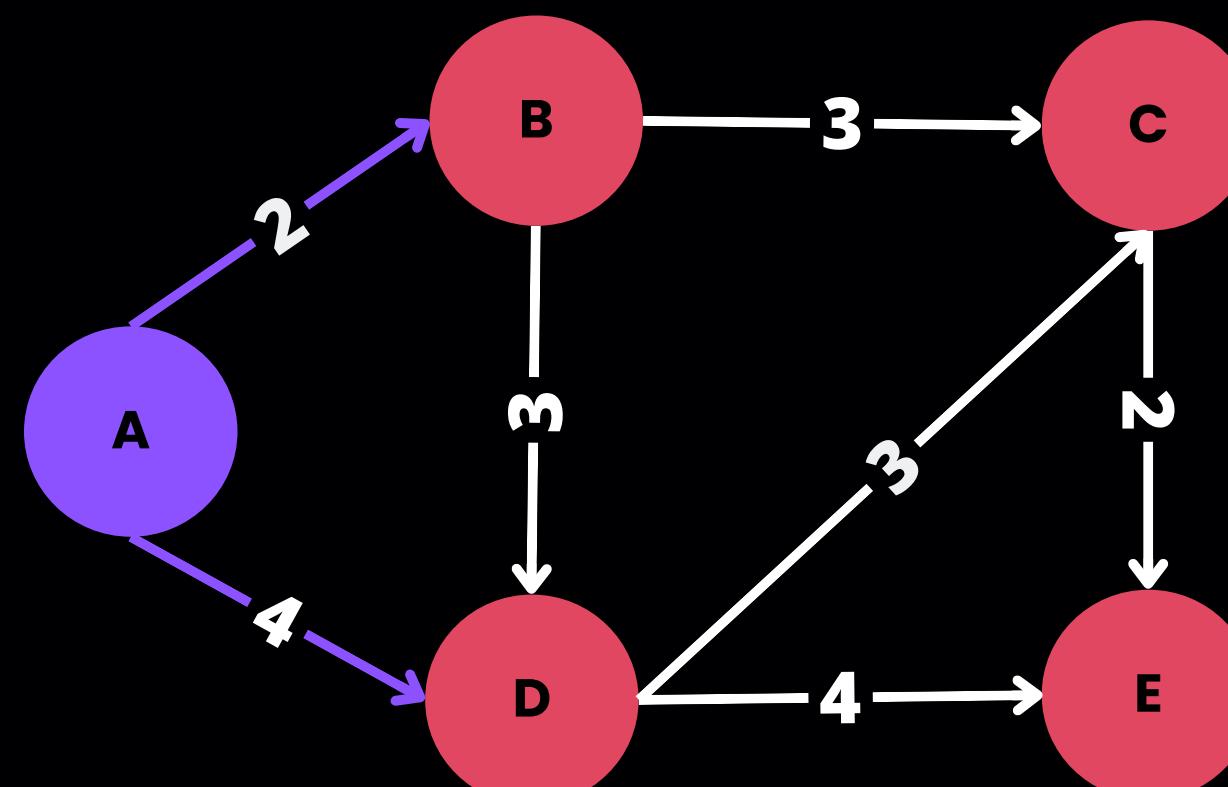


ALGORYTM DIJKSTRY

W	C	P
A	0	-
B	INF	-
C	INF	-
D	INF	-
E	INF	-



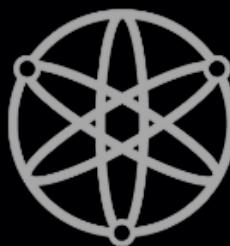
ALGORYTM DIJKSTRY



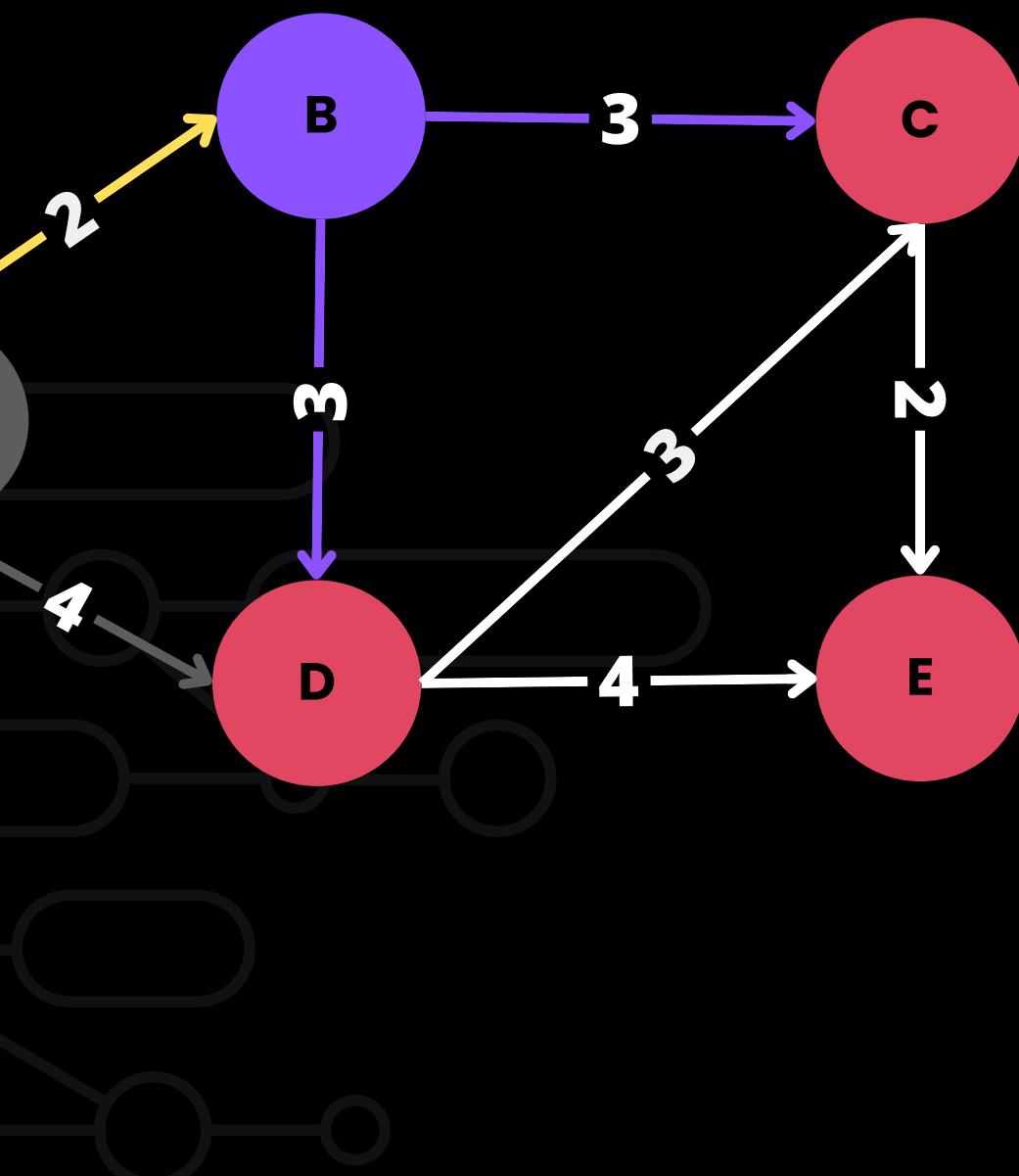
W	C	P
A	0	-
B	INF	-
C	INF	-
D	INF	-
E	INF	-

B: $\text{INF} > 0 + 2$
D: $\text{INF} > 0 + 4$

W	C	P
A	0	-
B	2	A
C	INF	-
D	4	A
E	INF	-



ALGORYTM DIJKSTRY

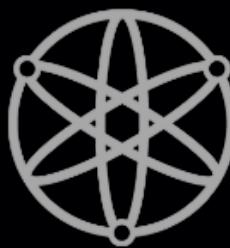


W	C	P
A	0	-
B	2	A
C	INF	-
D	4	A
E	INF	-

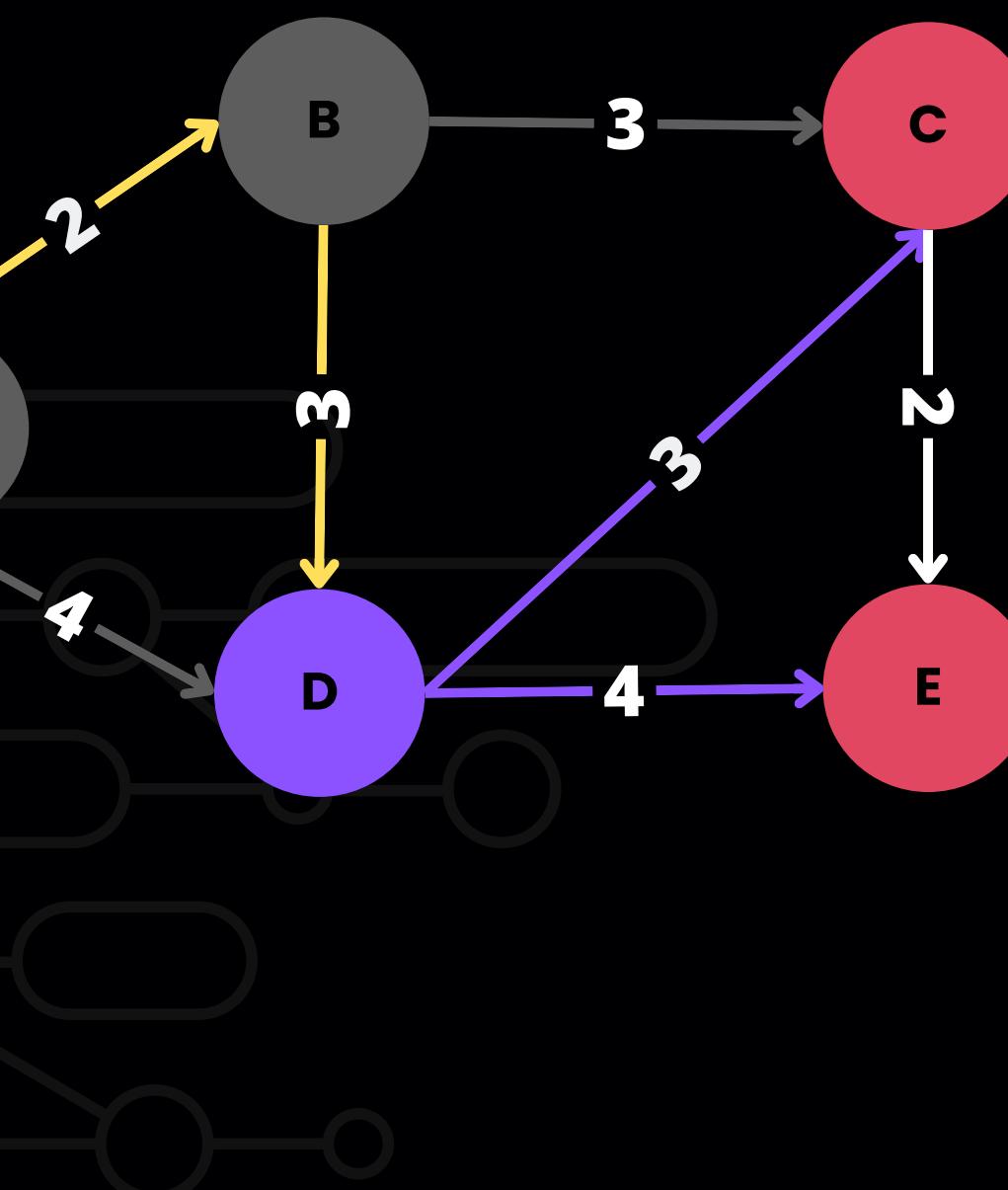
C: $INF > 3 + 2$

D: $2+3 = 5 > 4$

W	C	P
A	0	-
B	2	A
C	5	B
D	4	A
E	INF	-



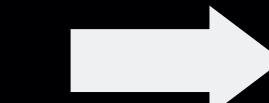
ALGORYTM DIJKSTRY



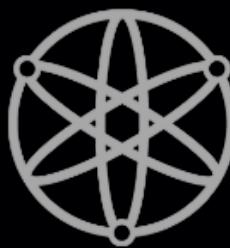
W	C	P
A	0	-
B	2	A
C	5	B
D	4	A
E	INF	-

E: $INF < 4 + 4 = 8$

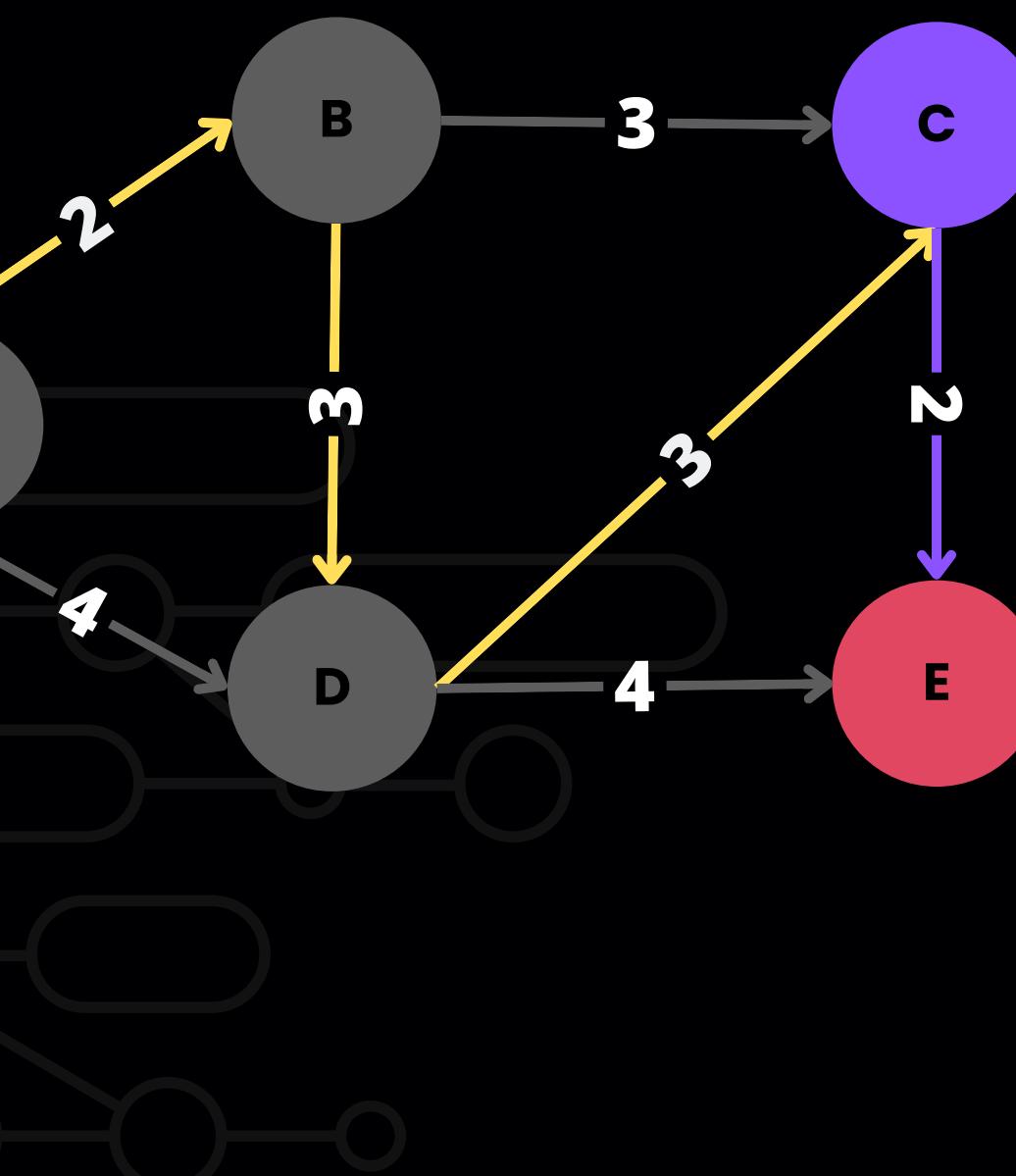
C: $4 + 3 = 7 > 5$



W	C	P
A	0	-
B	2	A
C	5	B
D	4	A
E	8	D



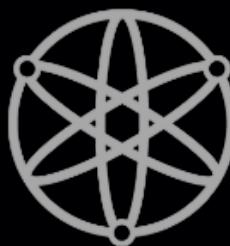
ALGORYTM DIJKSTRY



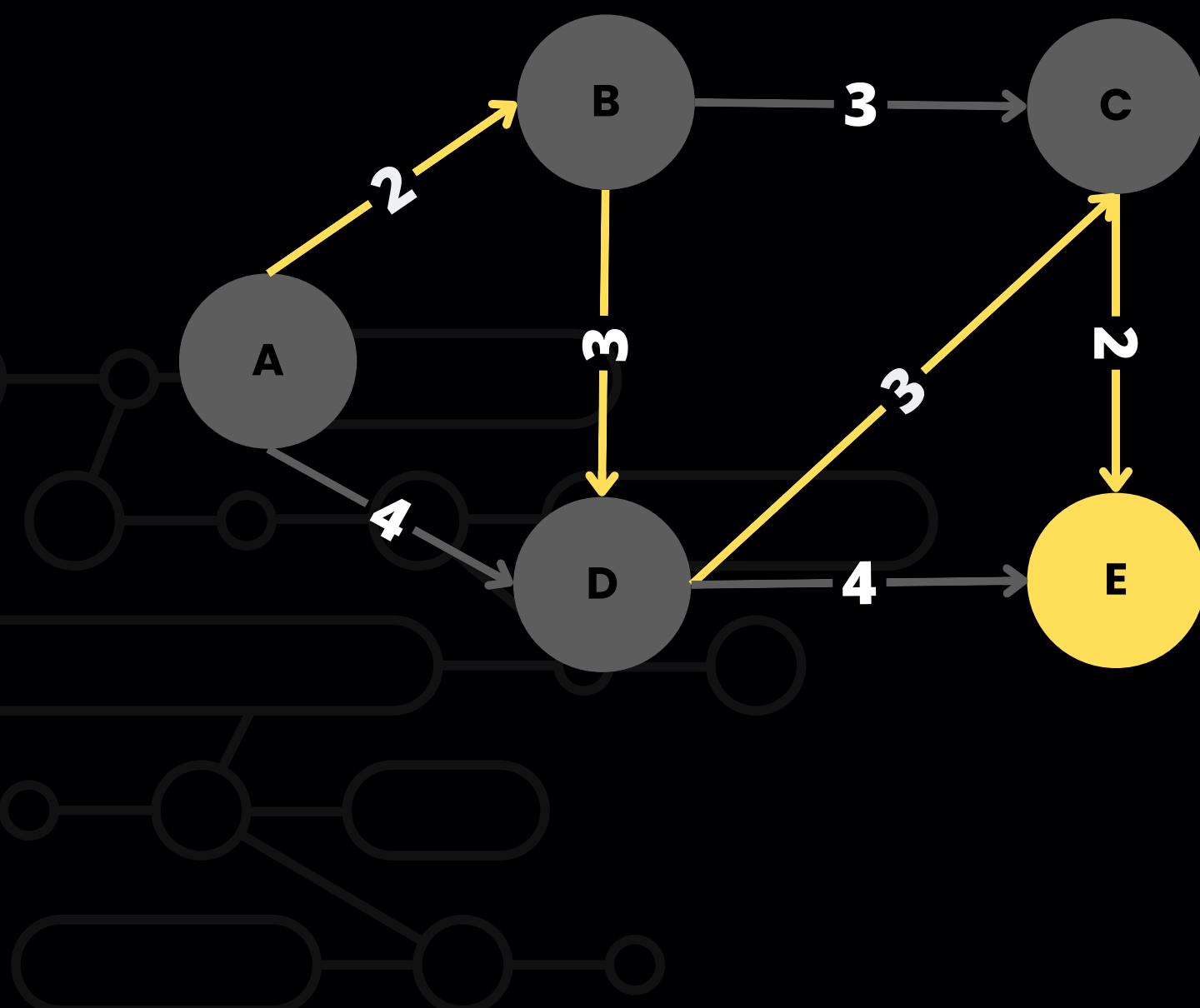
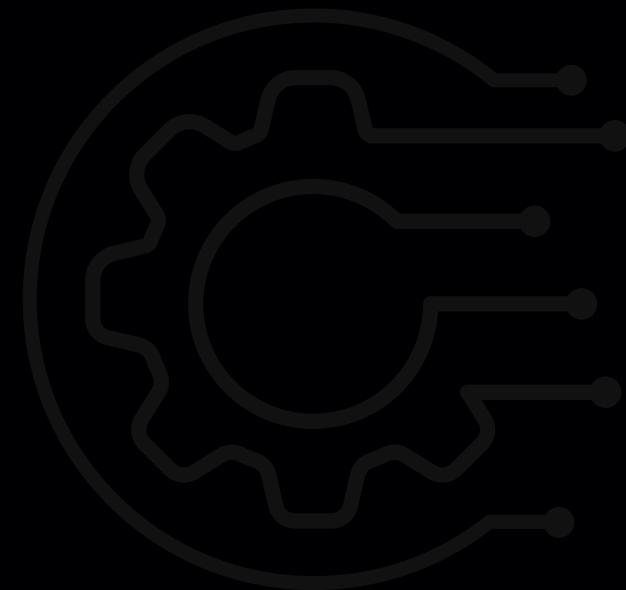
W	C	P
A	0	-
B	2	A
C	5	B
D	4	A
E	8	D

E: $5 + 2 = 7 < 8$

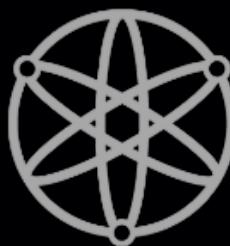
W	C	P
A	0	-
B	2	A
C	5	B
D	4	A
E	7	C



ALGORYTM DIJKSTRY

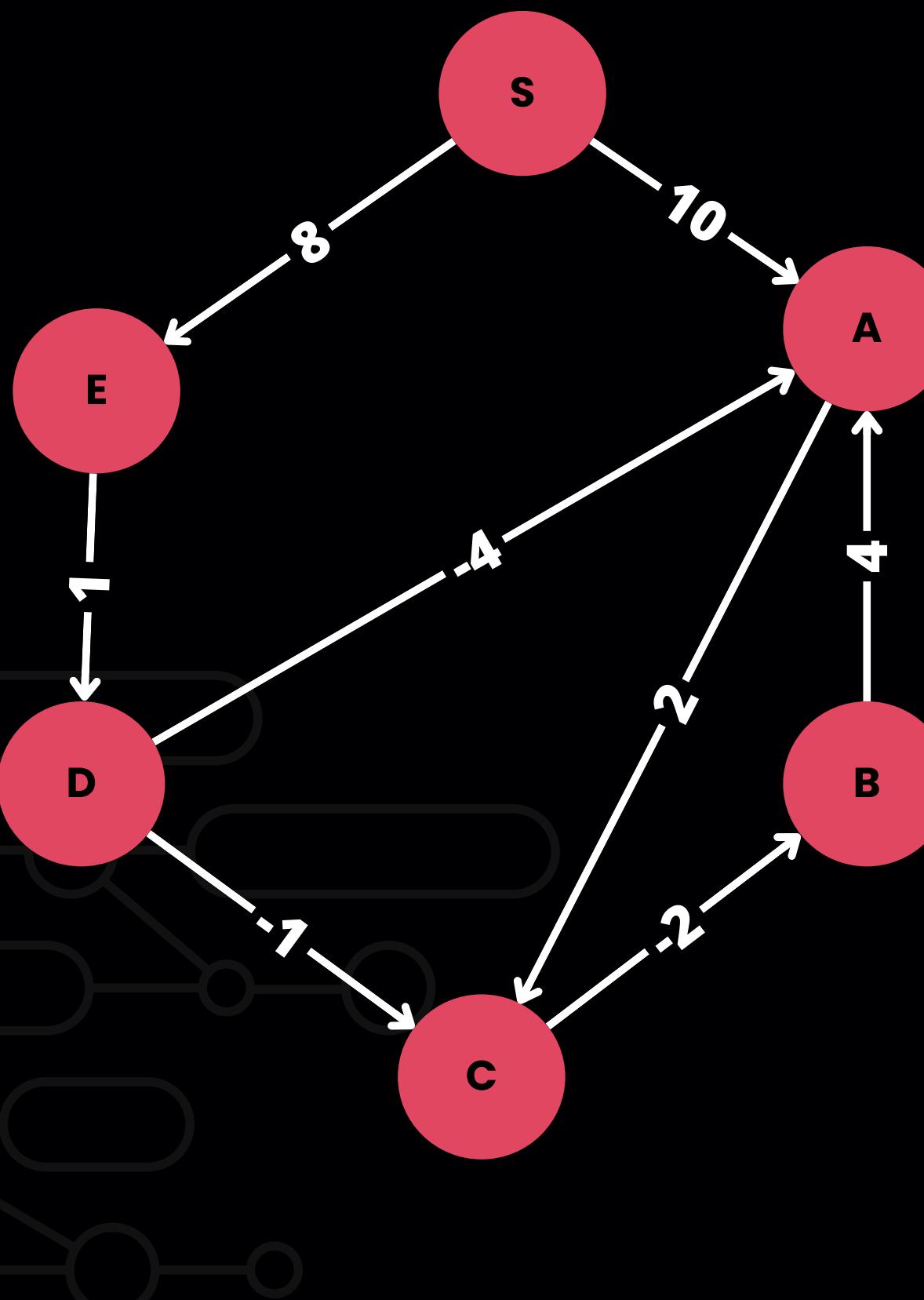


W	C	P
A	0	-
B	2	A
C	5	B
D	4	A
E	7	C



Najkrótsza ścieżka z jednego węzła do wszystkich węzłów, dozwolone wagi ujemne.

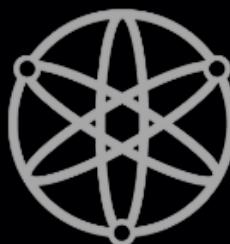
ALGORYTM BELLMANA-FORDA



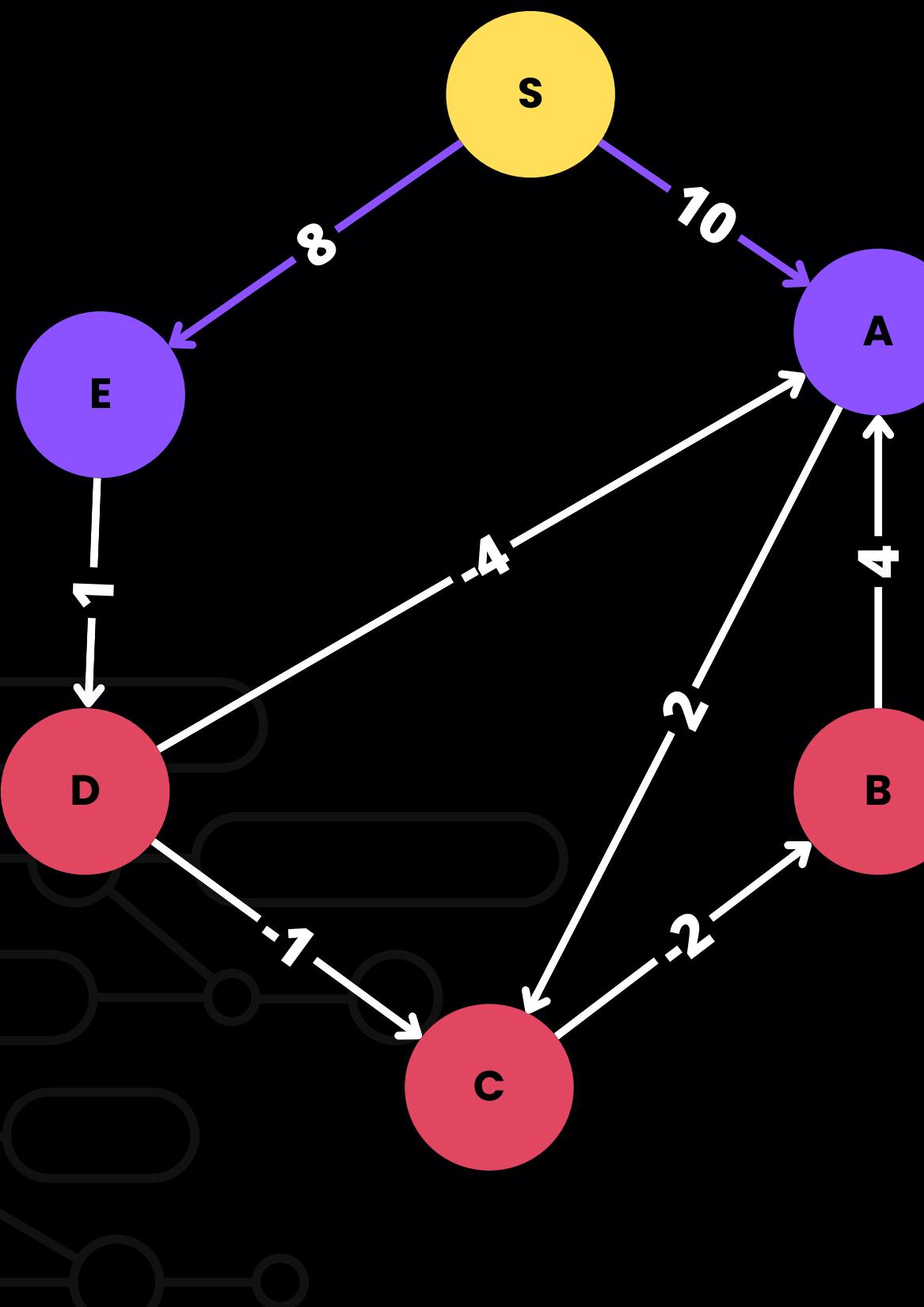
W	C
S	0
A	INF
B	INF
C	INF
D	INF
E	INF

1

ITERACJA



ALGORYTM BELLMANA-FORDA

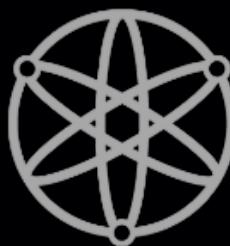


→

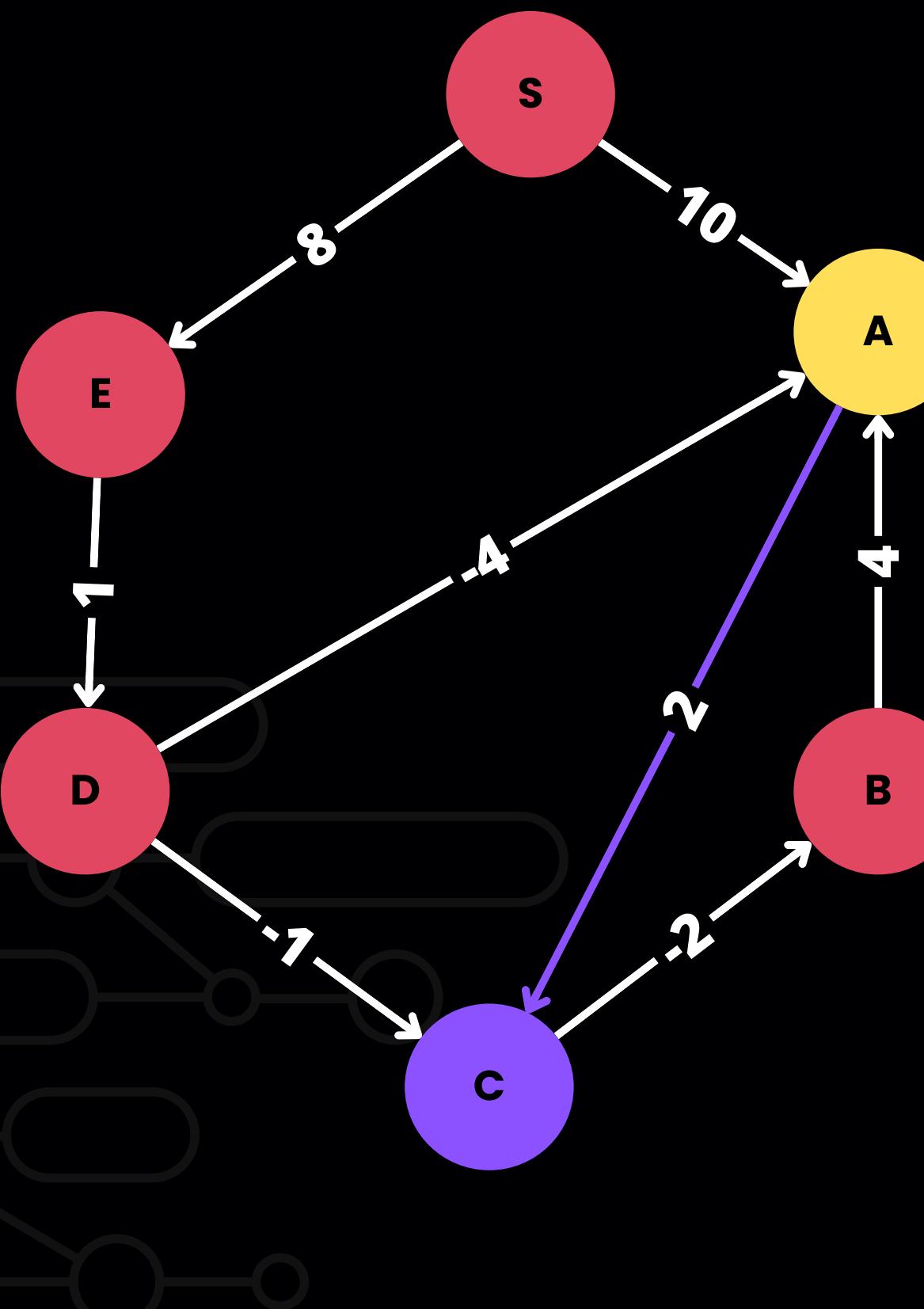
W	C
S	0
A	10
B	INF
C	INF
D	INF
E	8

ITERACJA

1



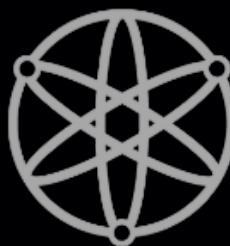
ALGORYTM BELLMANA-FORDA



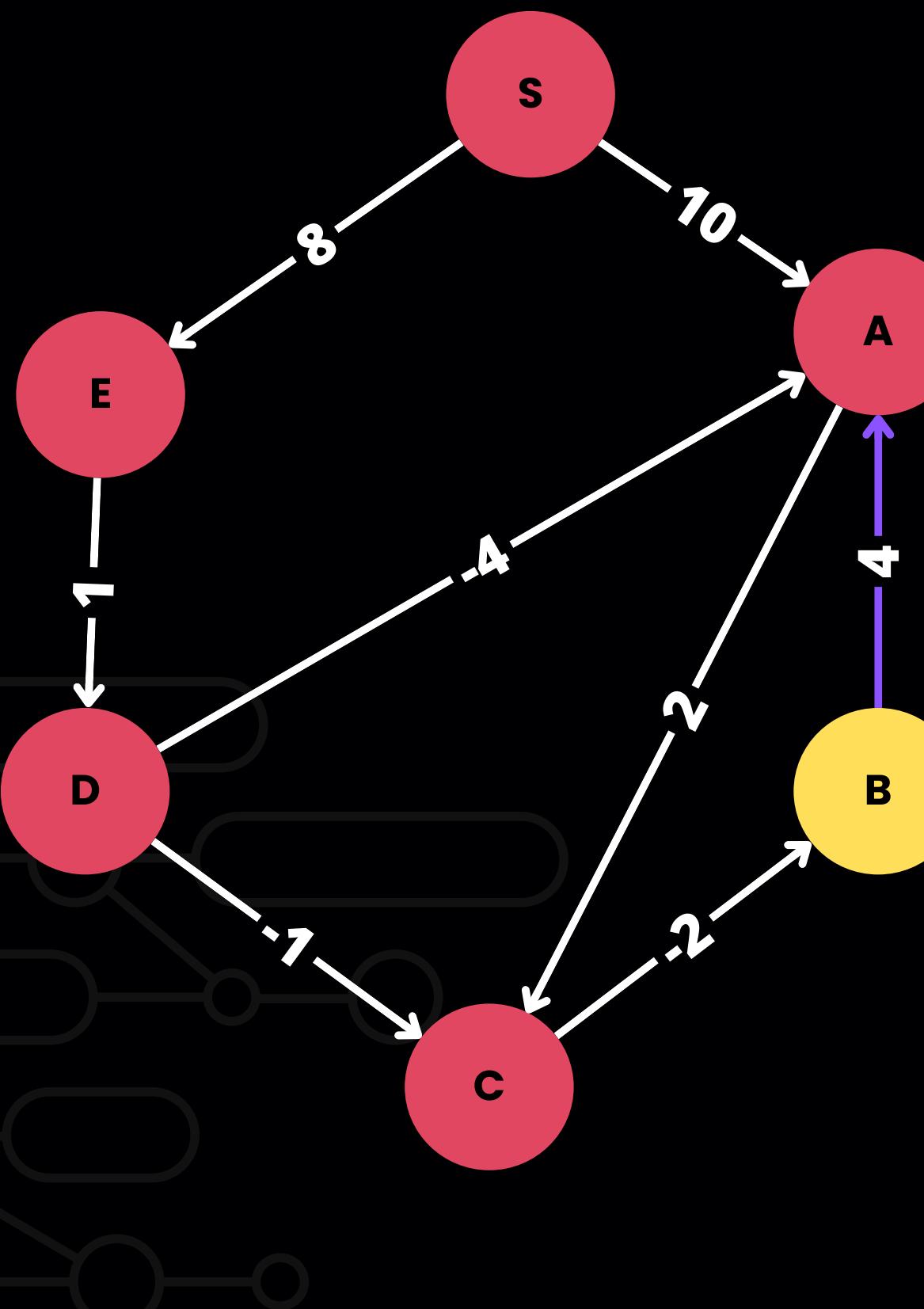
W	C
S	0
A	10
B	INF
C	12
D	INF
E	8

ITERACJA

1



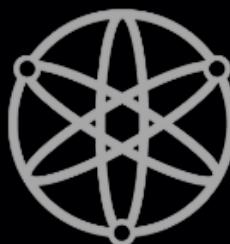
ALGORYTM BELLMANA-FORDA



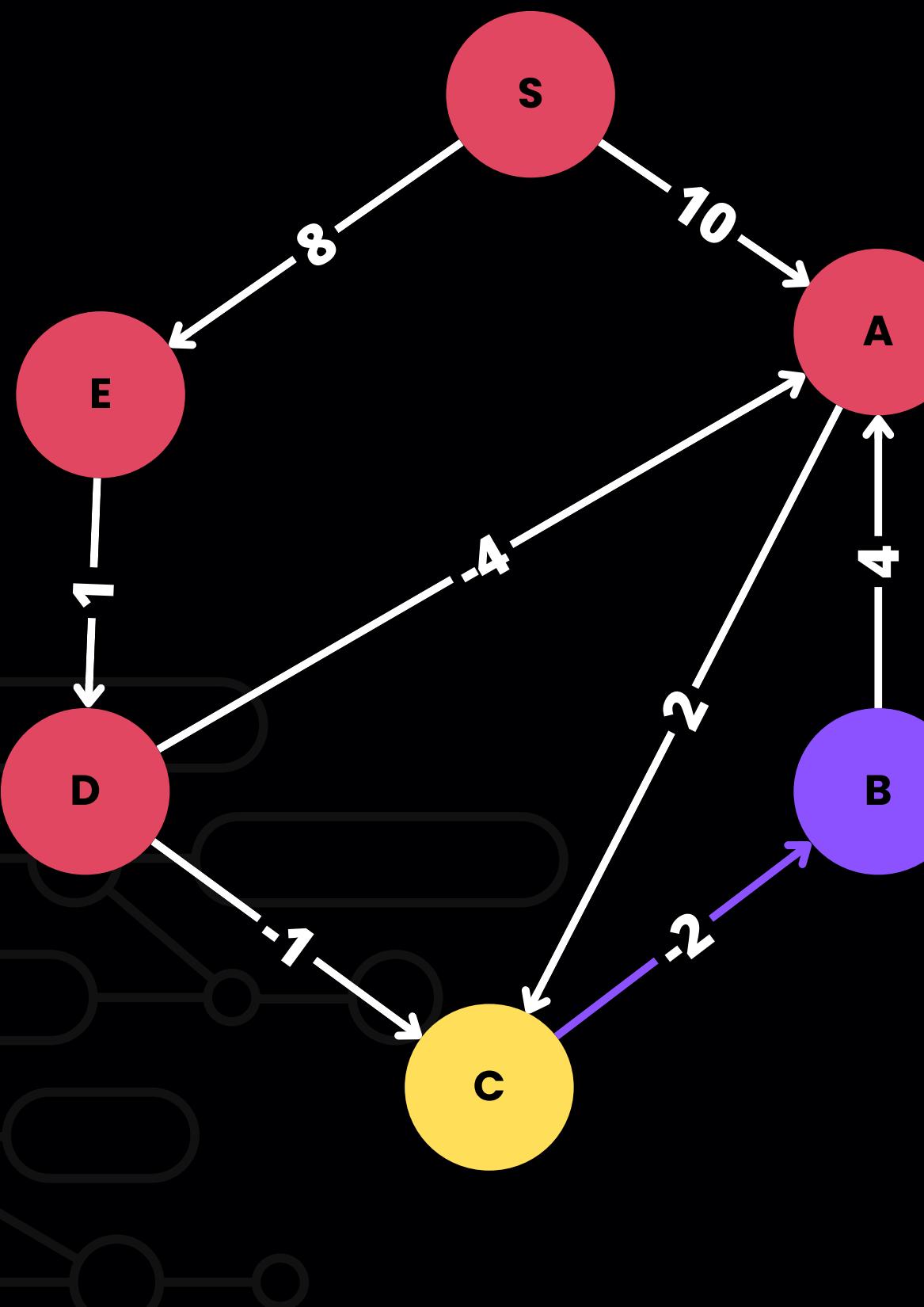
W	C
S	0
A	10
B	INF
C	12
D	INF
E	8

ITERACJA

1



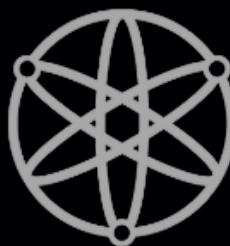
ALGORYTM BELLMANA-FORDA



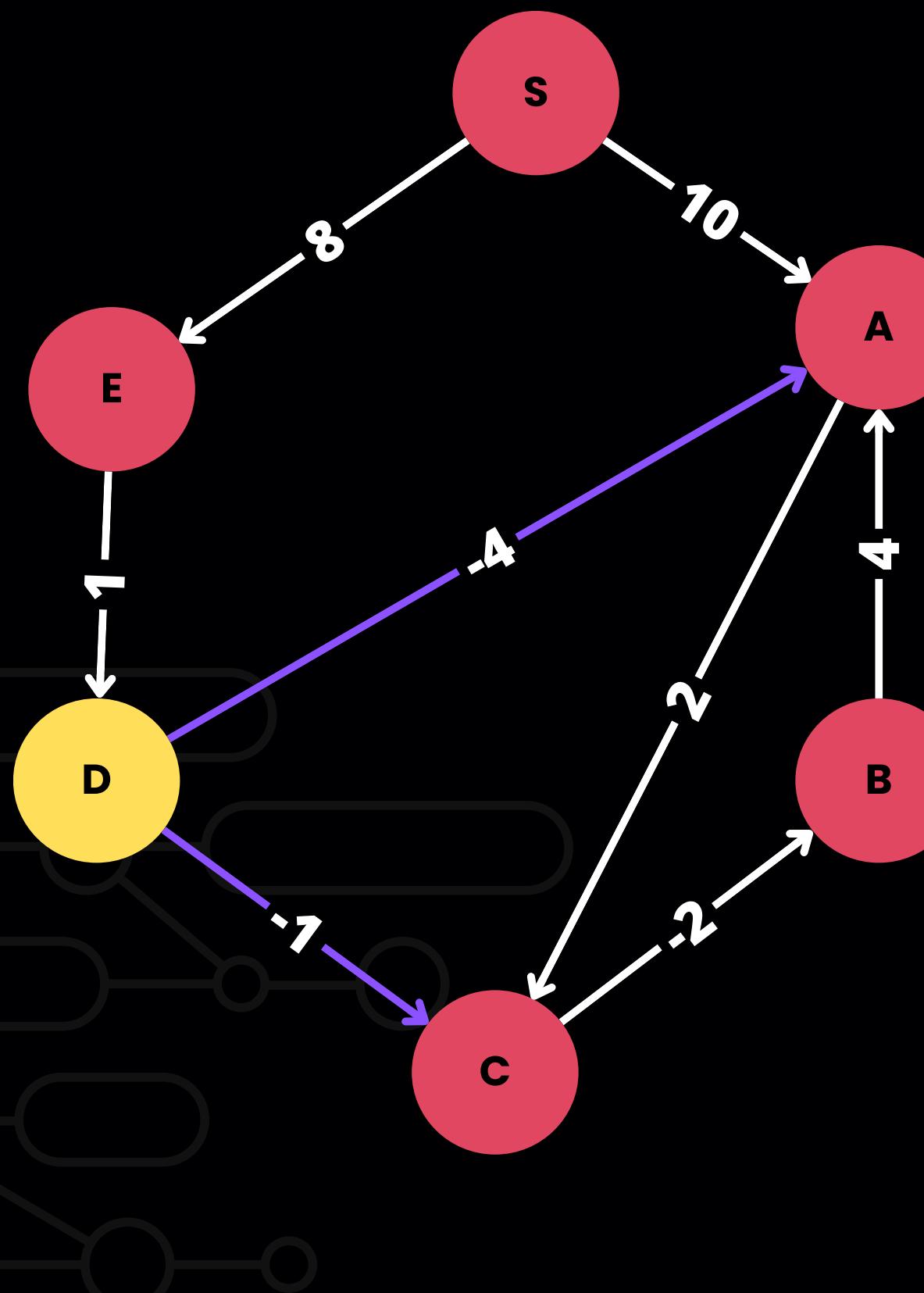
W	C
S	0
A	10
B	10
C	12
D	INF
E	8

ITERACJA

1



ALGORYTM BELLMANA-FORDA

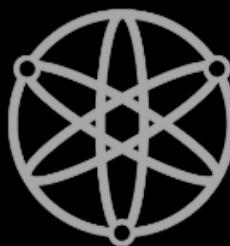


W	C
S	0
A	10
B	10
C	12
D	INF
E	8

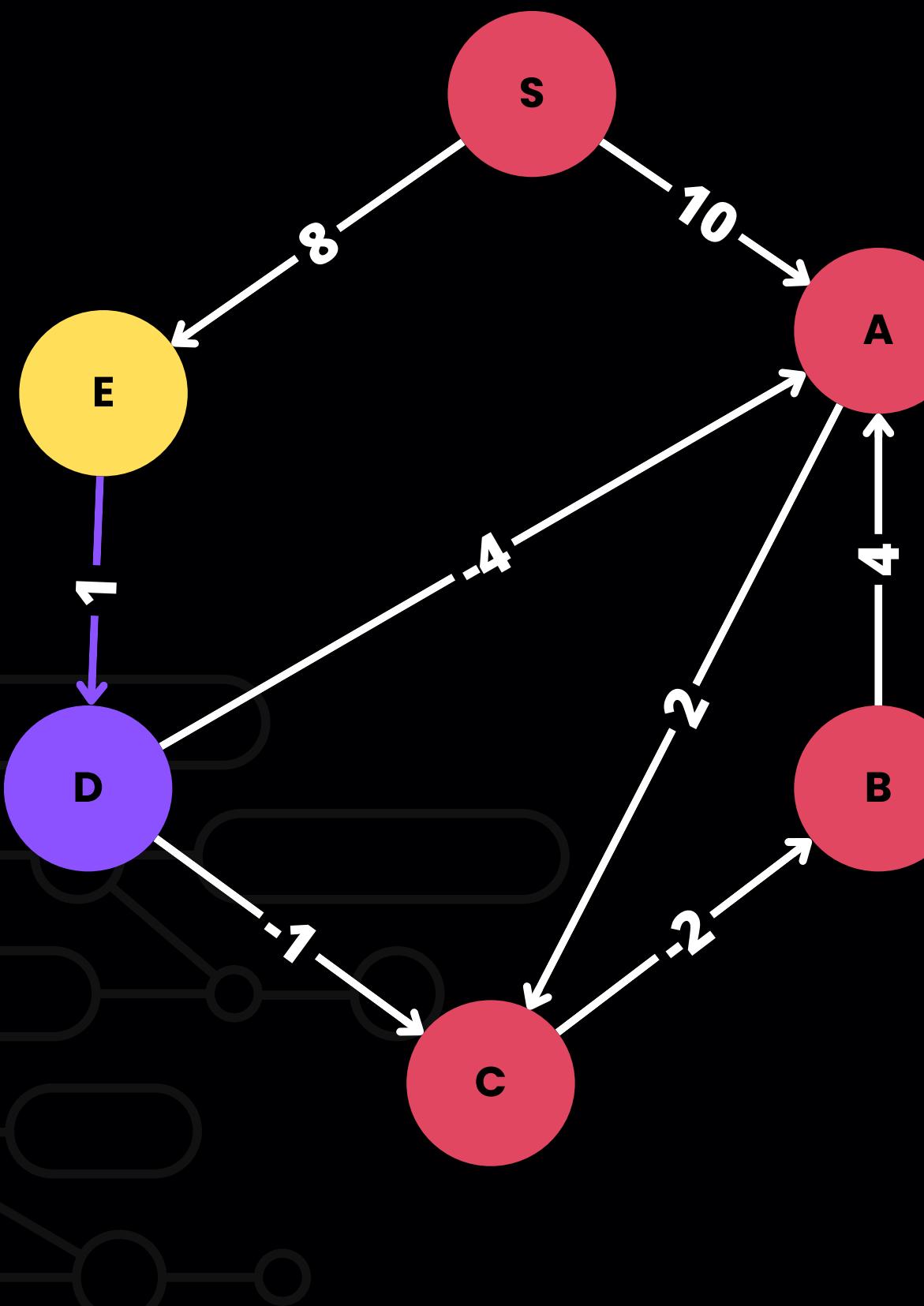


ITERACJA

1



ALGORYTM BELLMANA-FORDA

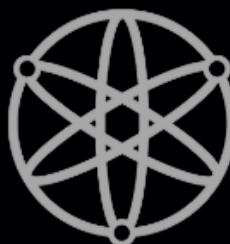


W	C
S	0
A	10
B	10
C	12
D	9
E	8

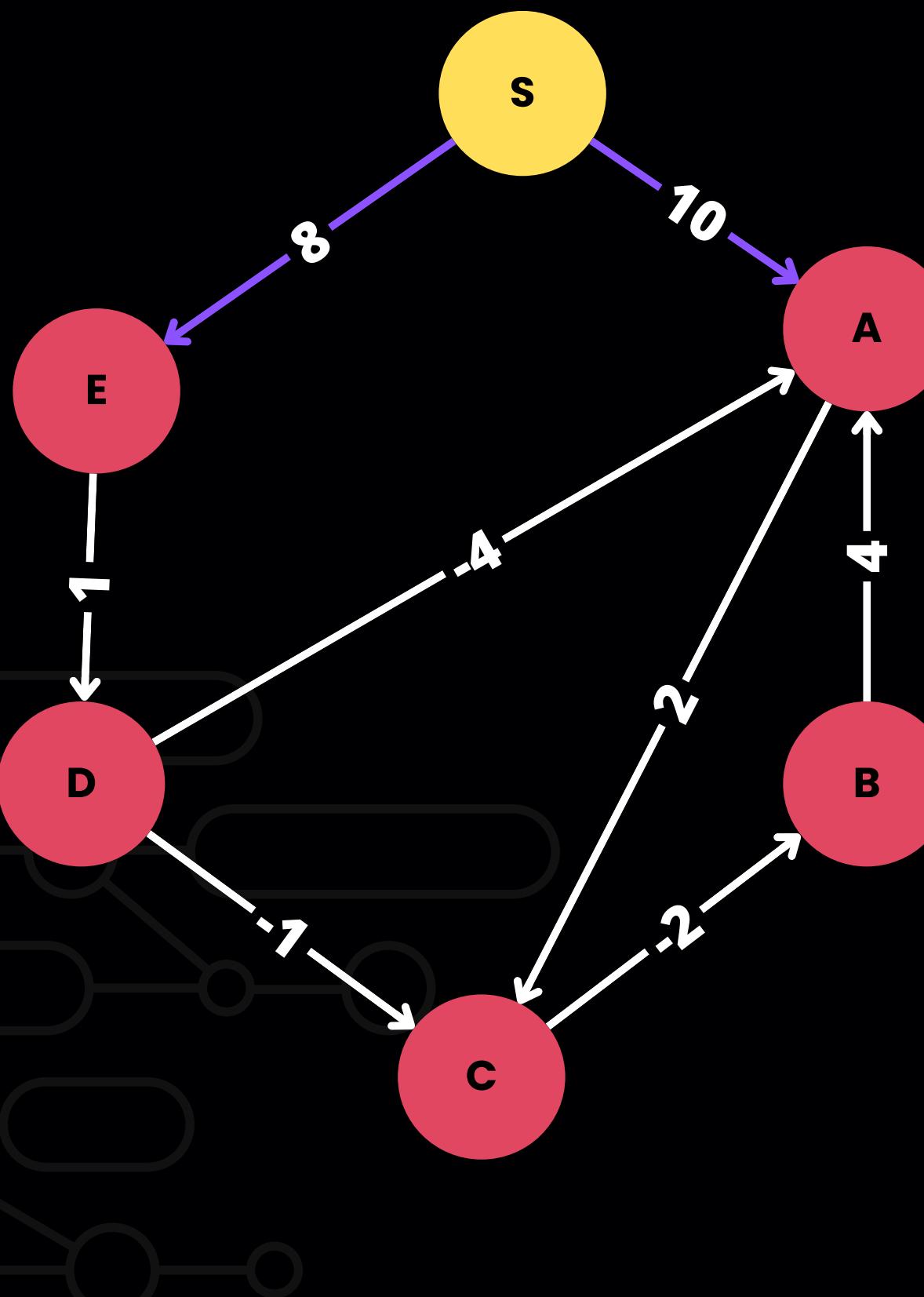


ITERACJA

1



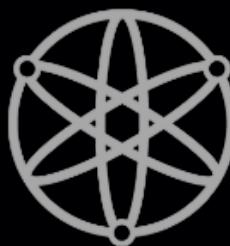
ALGORYTM BELLMANA-FORDA



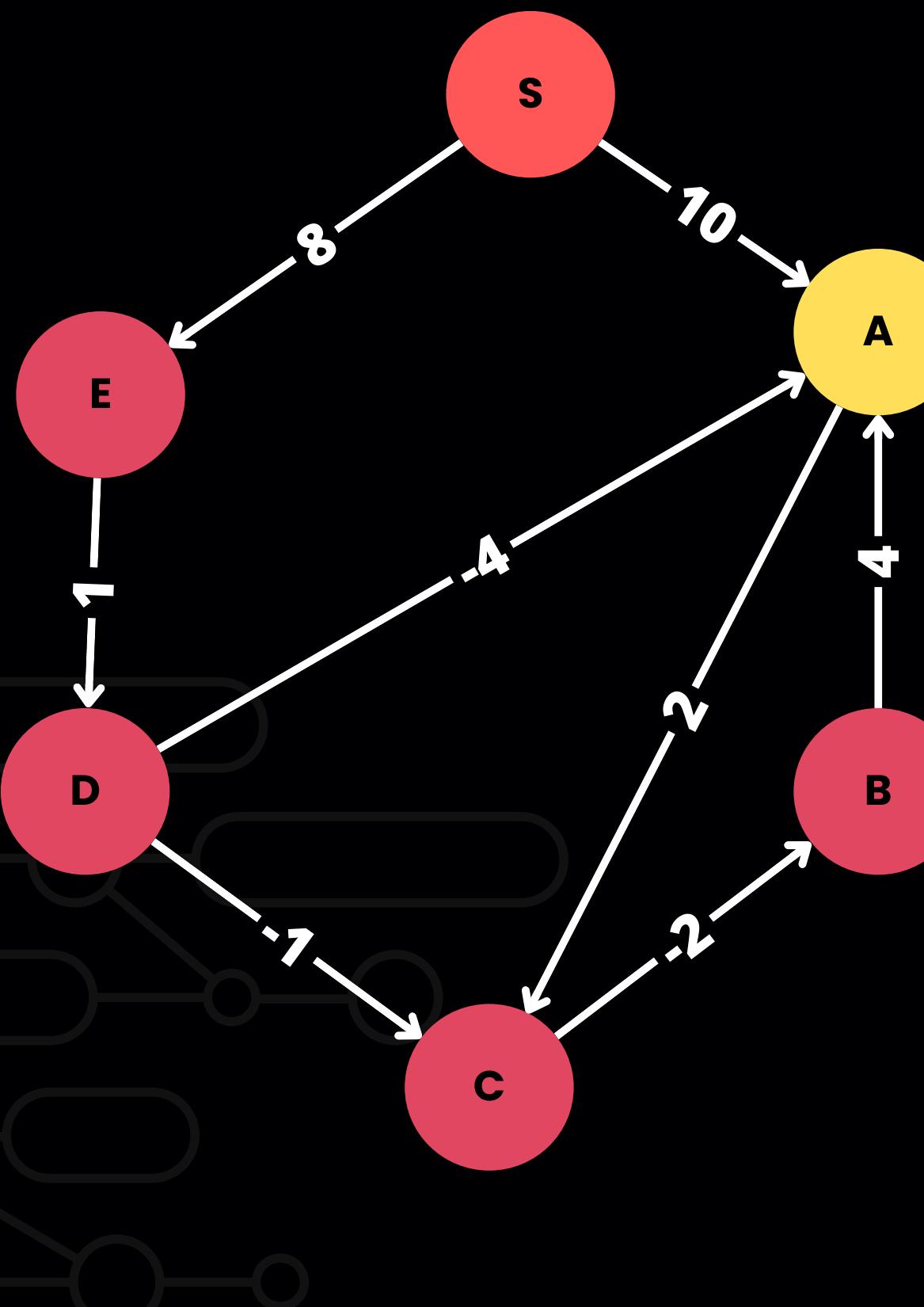
W	C
S	0
A	10
B	10
C	12
D	9
E	8

2

ITERACJA



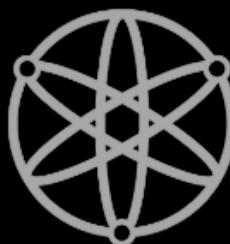
ALGORYTM BELLMANA-FORDA



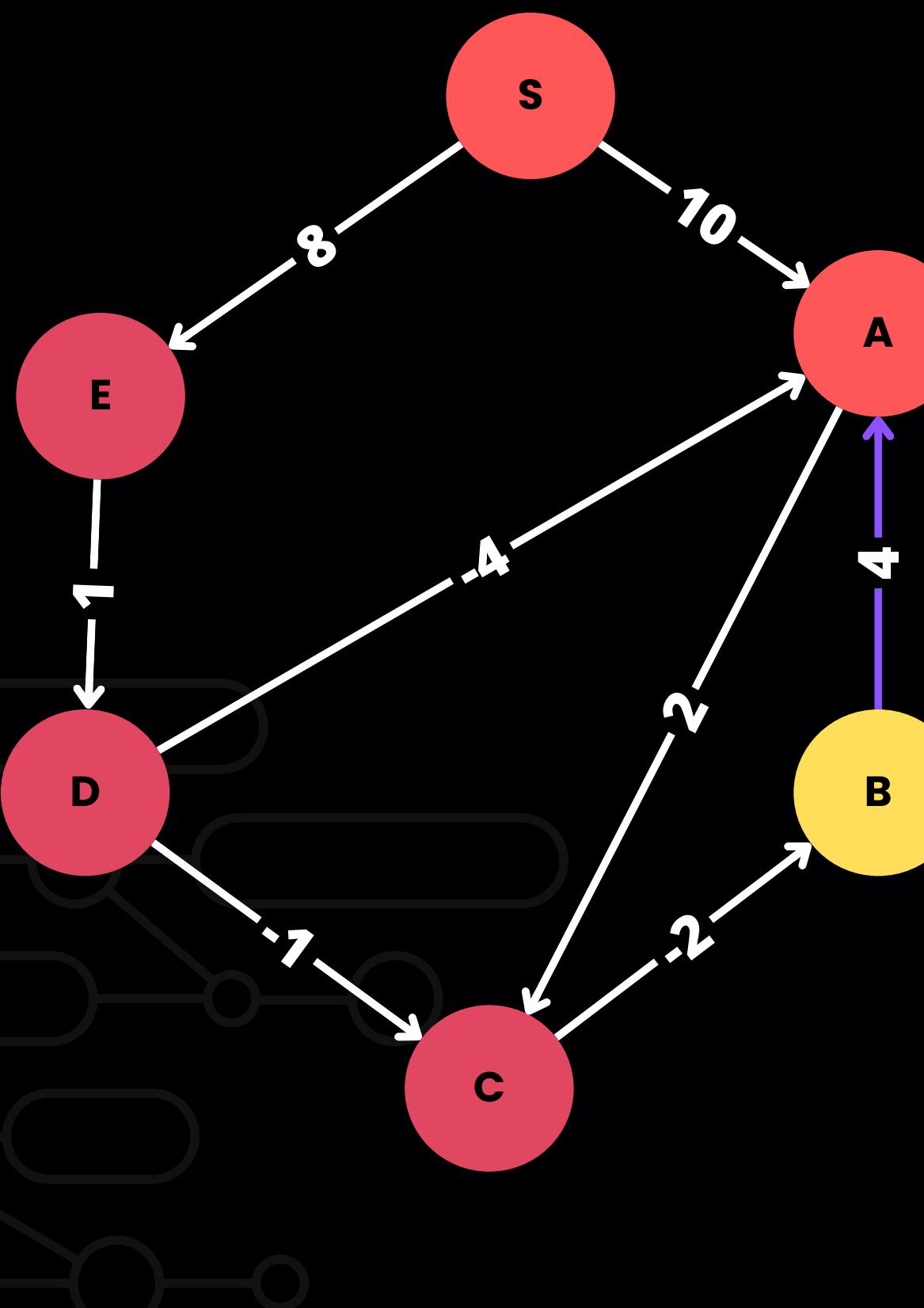
W	C
S	0
A	10
B	10
C	12
D	9
E	8

2

ITERACJA



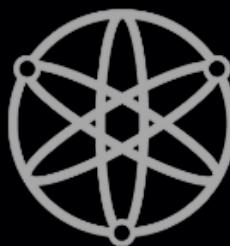
ALGORYTM BELLMANA-FORDA



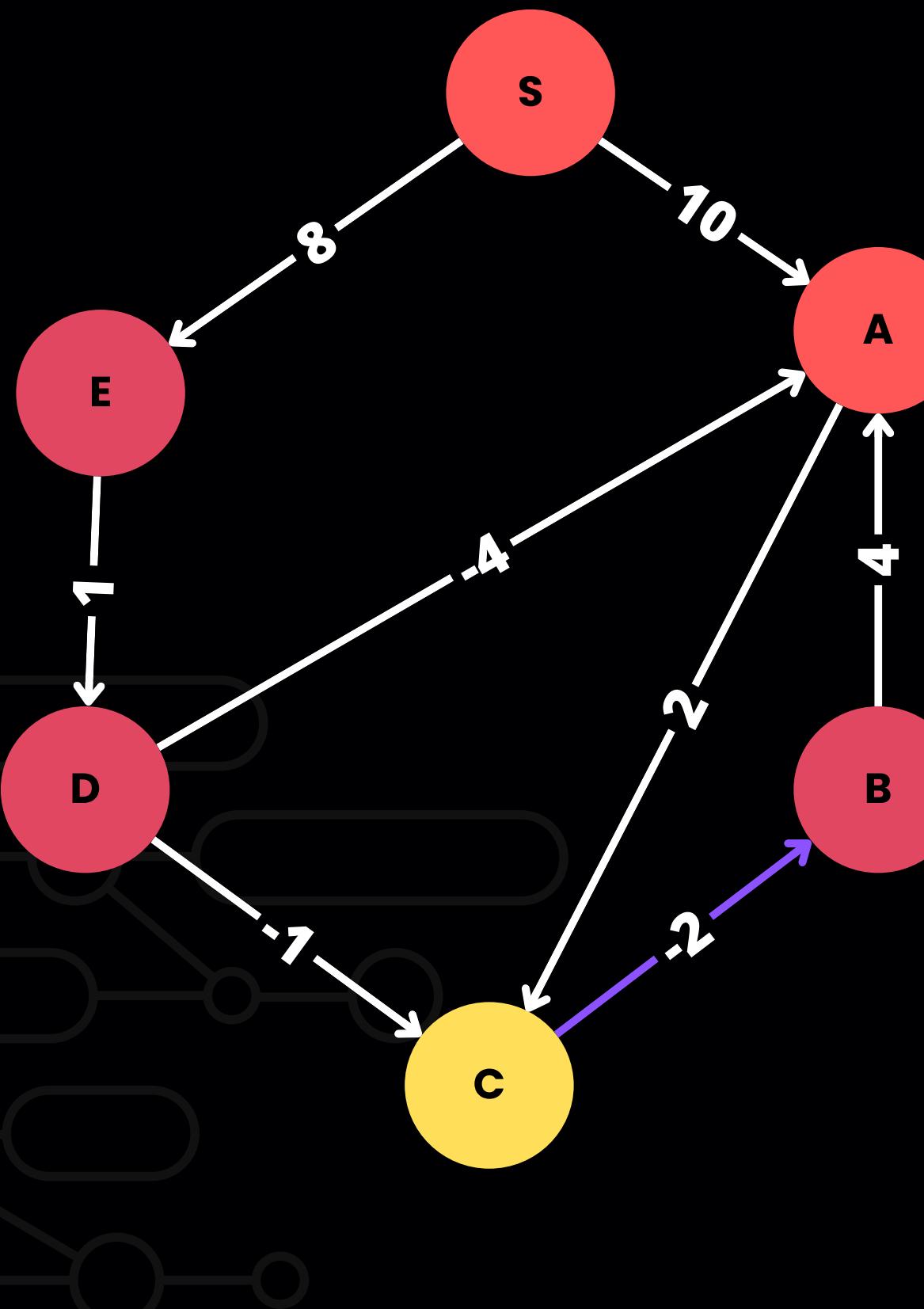
W	C
S	0
A	10
B	10
C	12
D	9
E	8

2

ITERACJA



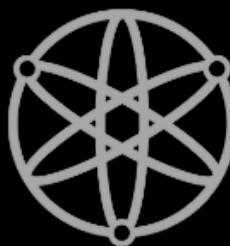
ALGORYTM BELLMANA-FORDA



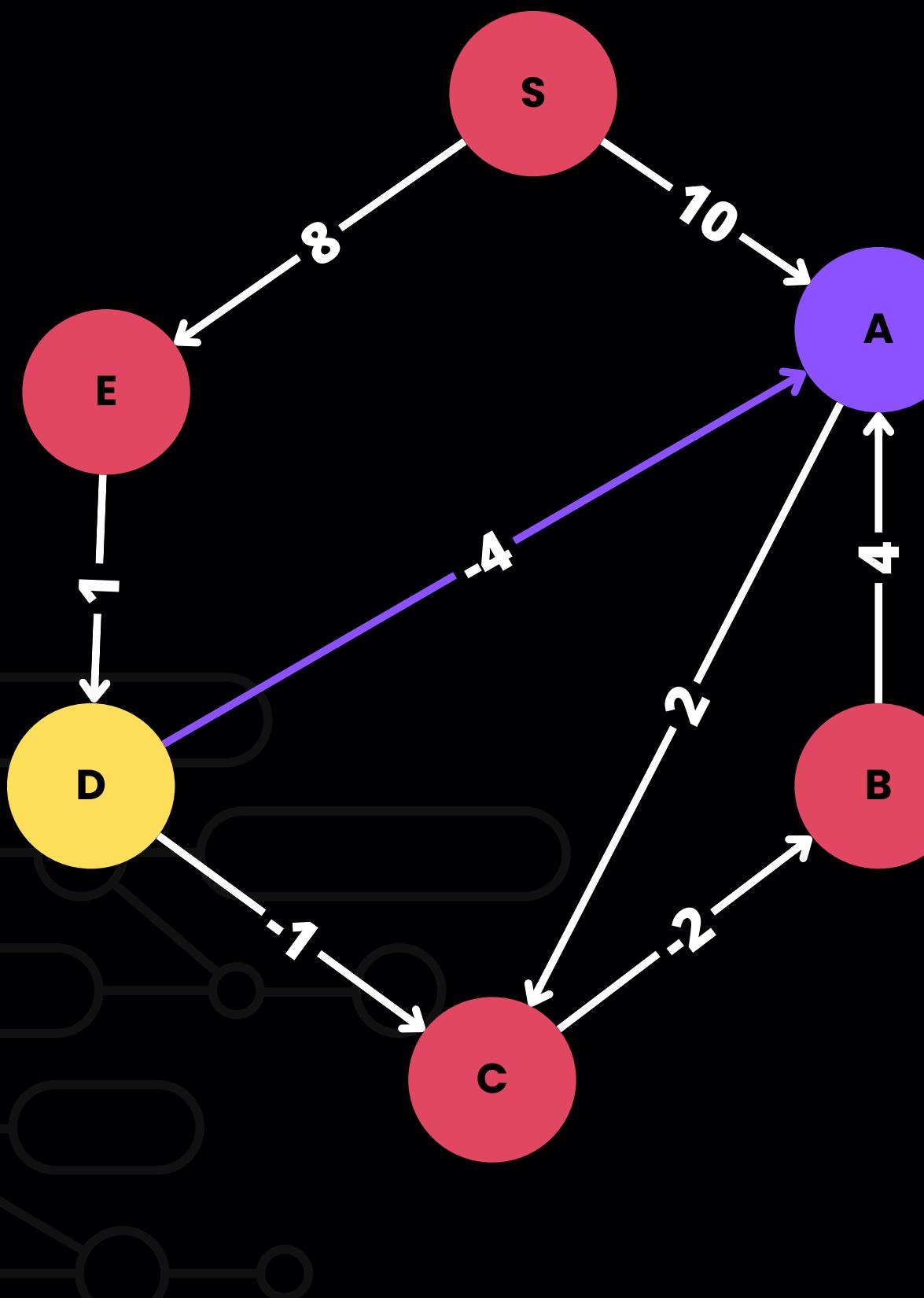
W	C
S	0
A	10
B	10
C	12
D	9
E	8

2

ITERACJA



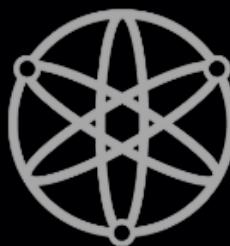
ALGORYTM BELLMANA-FORDA



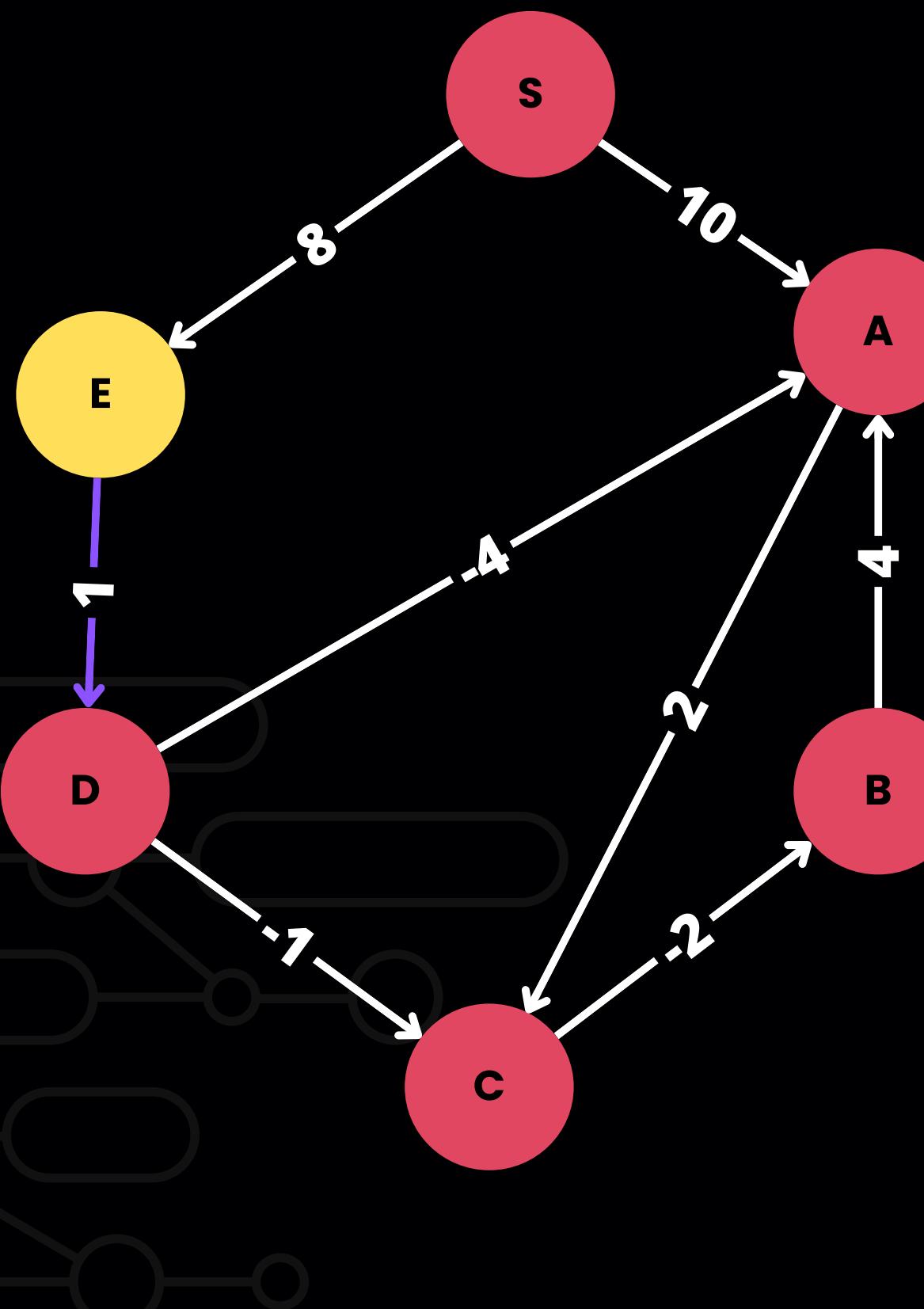
W	C
S	0
A	5
B	10
C	12
D	9
E	8

ITERACJA

2



ALGORYTM BELLMANA-FORDA

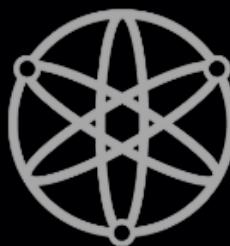


W	C
S	0
A	5
B	10
C	12
D	9
E	8



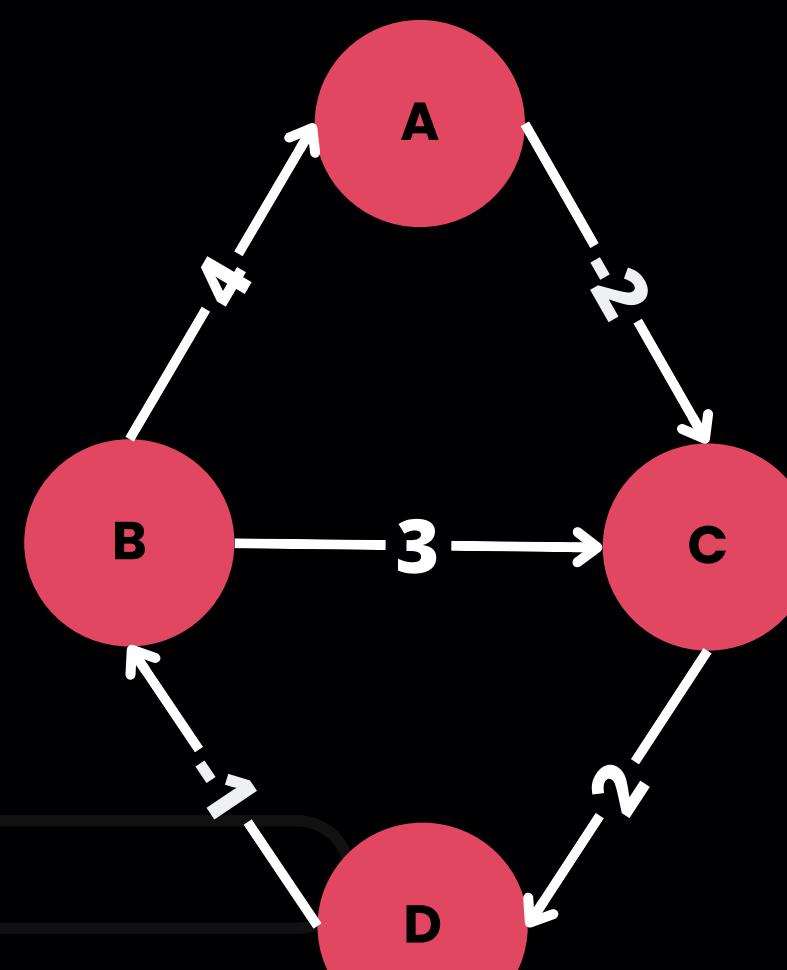
ITERACJA

2

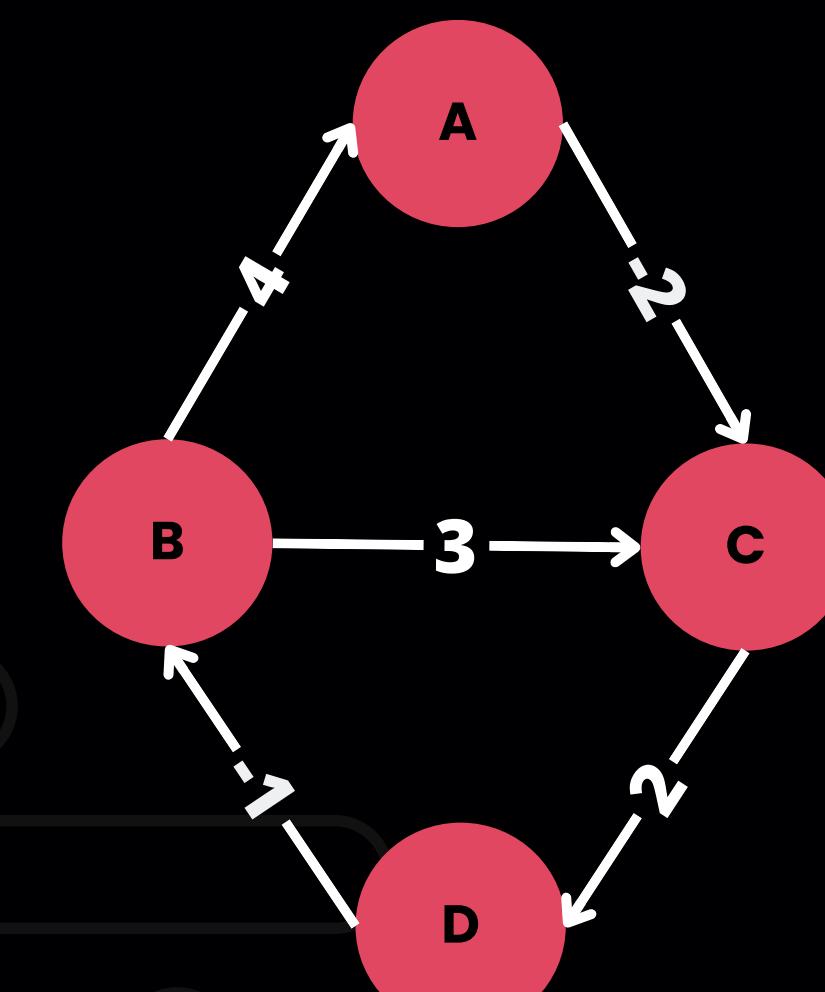
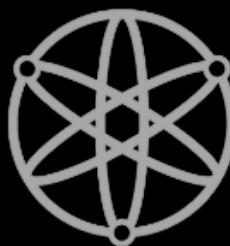


Najkrótsza ścieżka od jednego węzła do wszystkich węzłów, dozwolone wagi ujemne.

ALGORYTM FLOYDA – WARSHALLA

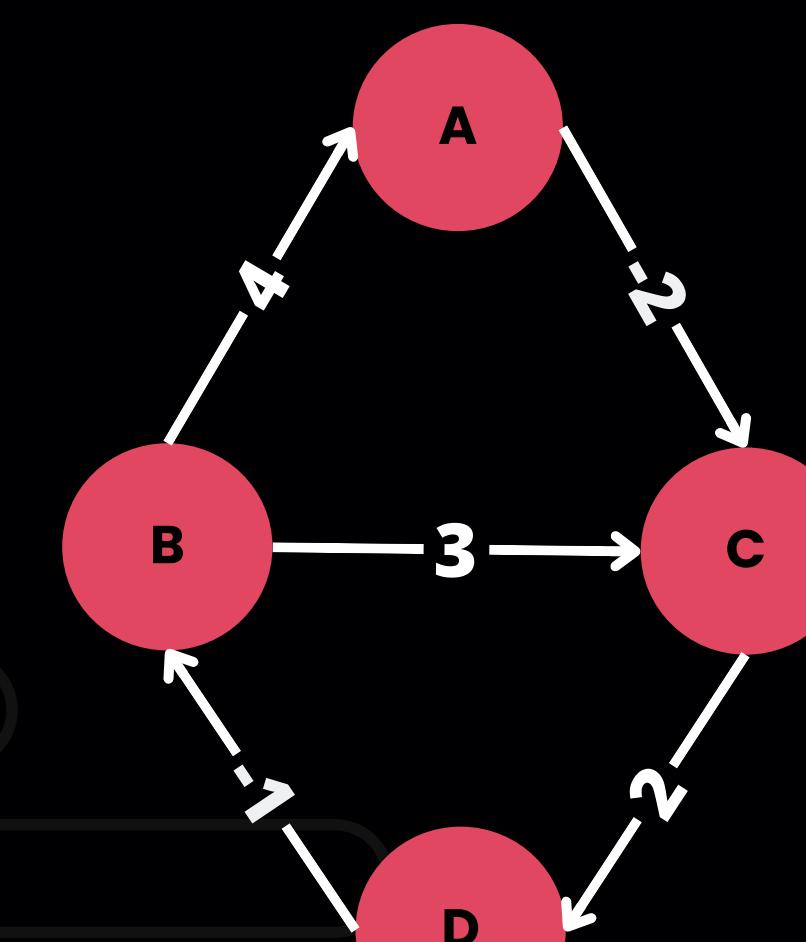
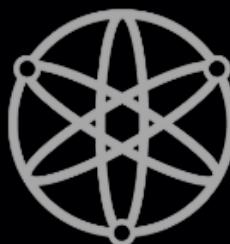


	A	B	C	D
A				
B				
C				
D				



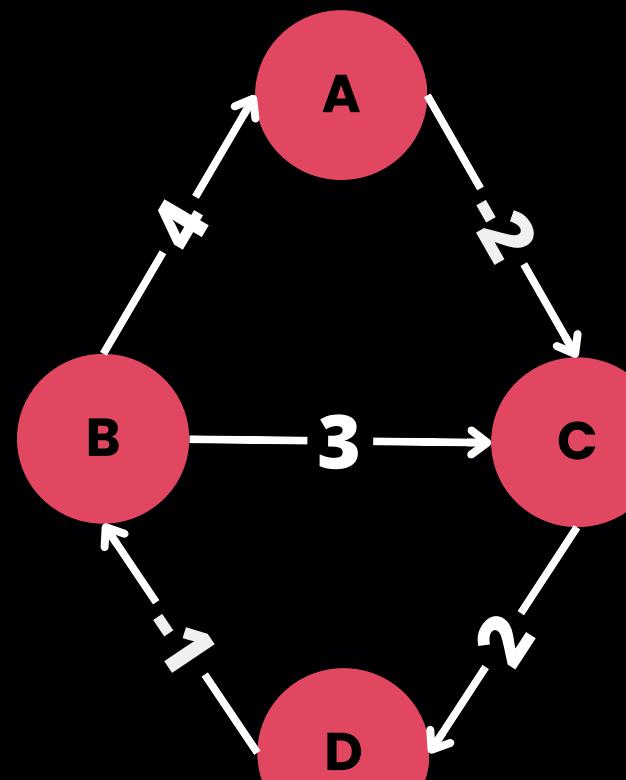
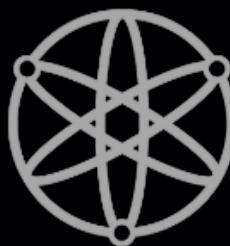
ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0			
B		0		
C			0	
D				0



ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0	INF	-2	INF
B	4	0	3	INF
C	INF	INF	0	2
D	INF	-1	INF	0



K: A B C D

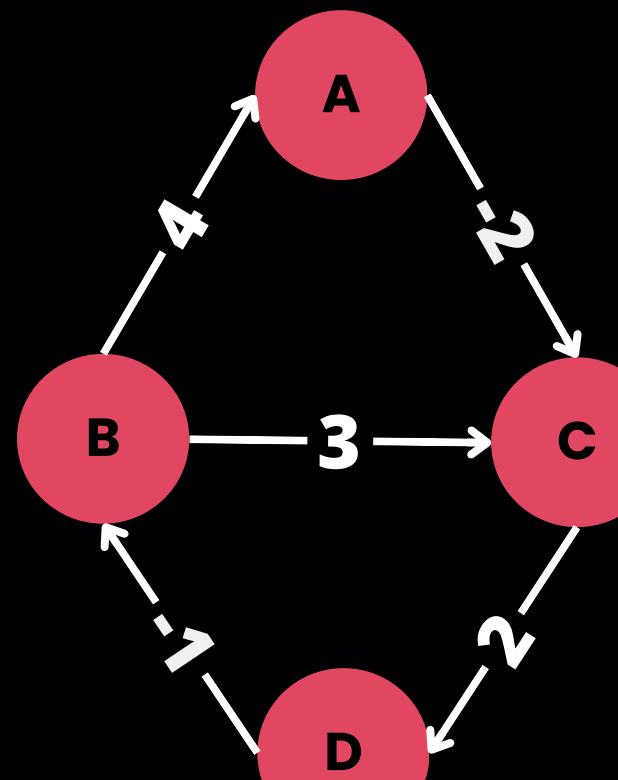
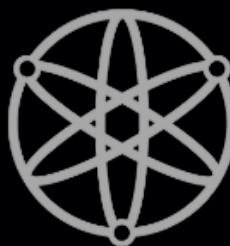
I: A B C D

J: A B C D

$$\text{Odl}[i][j] > \text{Odl}[i][j] + \text{Odl}[k][j] \rightarrow \text{Odl}[i][j] = \text{Odl}[i][j] + \text{Odl}[k][j]$$

ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0	INF	-2	INF
B	4	0	3	INF
C	INF	INF	0	2
D	INF	-1	INF	0



K: **A B C D**

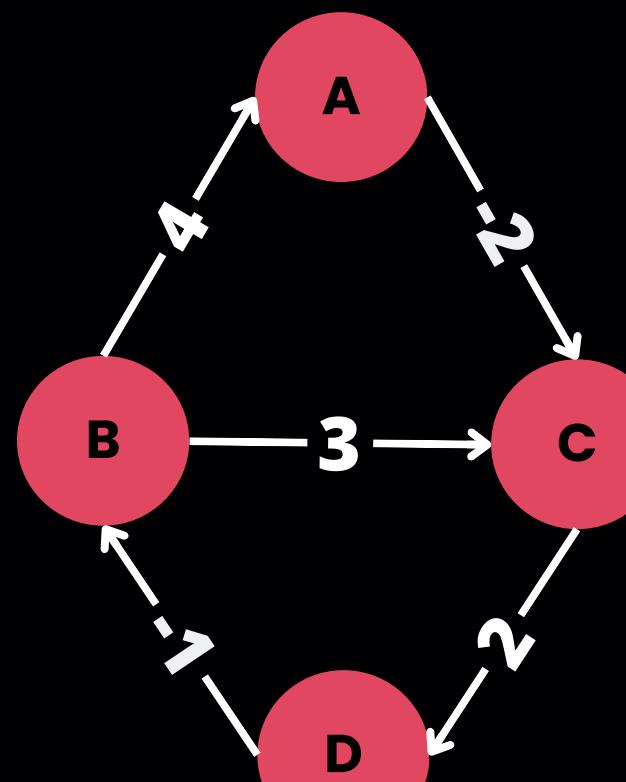
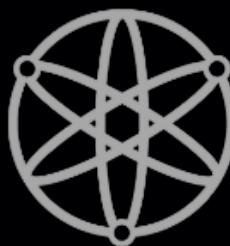
I: **A B C D**

J: **A B C D**

ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0	INF	-2	INF
B	4	0	3	INF
C	INF	INF	0	2
D	INF	-1	INF	0

$$\text{Odl}[A][A] \rightarrow \text{Odl}[A][A] + \text{Odl}[A][A] \rightarrow 0 \rightarrow 0 + 0 \rightarrow 0 \rightarrow 0$$



K: A B C D

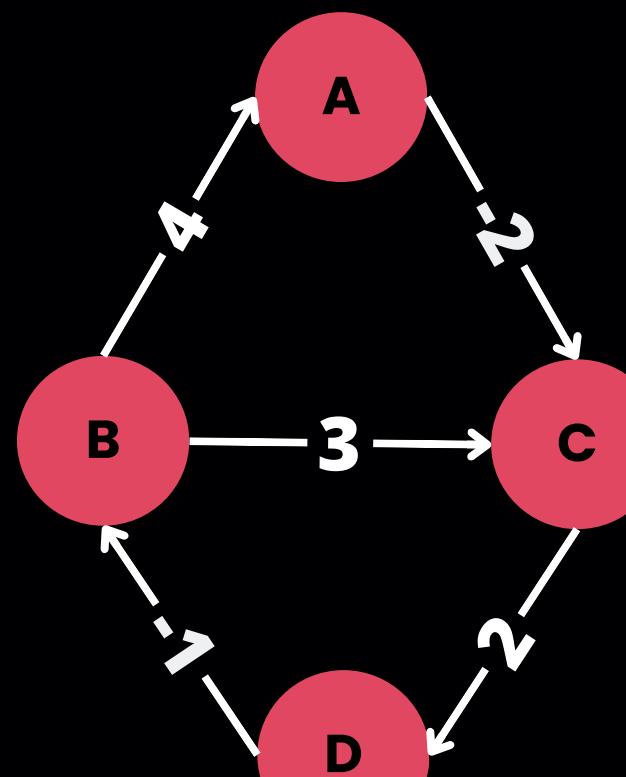
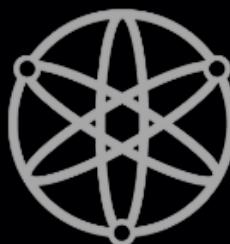
I: A B C D

J: A B C D

ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0	INF	-2	INF
B	4	0	3	INF
C	INF	INF	0	2
D	INF	-1	INF	0

$$\text{Odl}[A][B] > \text{Odl}[A][A] + \text{Odl}[A][B] \rightarrow \text{INF} > 0 + \text{INF} \rightarrow \text{INF} > \text{INF}$$



K: A B C D

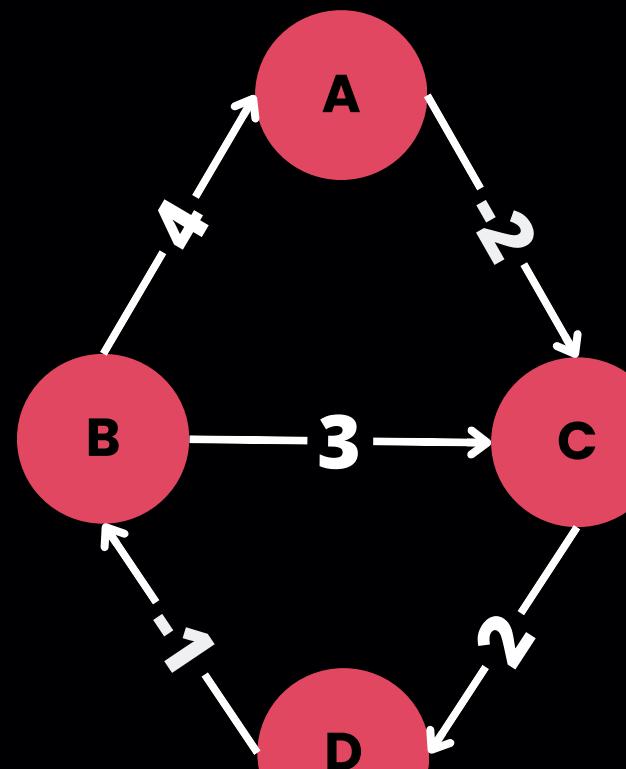
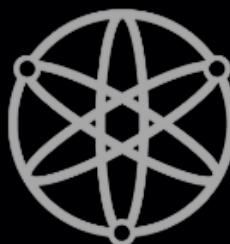
I: A B C D

J: A B C D

$$\text{Odl}[A][C] \rightarrow \text{Odl}[A][A] + \text{Odl}[A][C] \rightarrow -2 \rightarrow 0 + -2 \rightarrow -2 \rightarrow -2$$

ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0	INF	-2	INF
B	4	0	3	INF
C	INF	INF	0	2
D	INF	-1	INF	0



K: **A B C D**

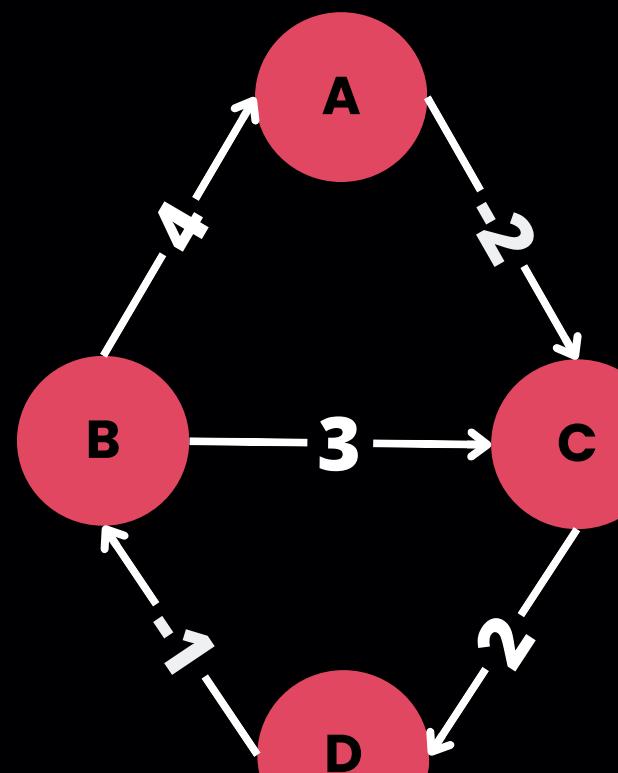
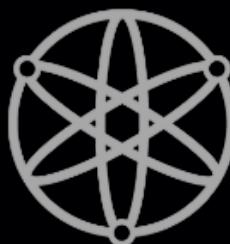
I: **A B C D**

J: **A B C D**

$$\text{Odl[B][c]} \rightarrow \text{Odl[B][A]} + \text{Odl[A][c]} \rightarrow 3 > 4 + -2 \rightarrow 3 > 2 \rightarrow \text{Odl[B][c]} = 2$$

ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0	INF	-2	INF
B	4	0	3	INF
C	INF	INF	0	2
D	INF	-1	INF	0



K: A **B C D**

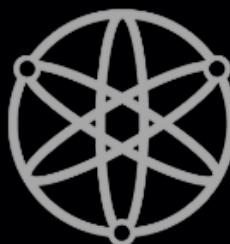
I: A **B C D**

J: **A B C D**

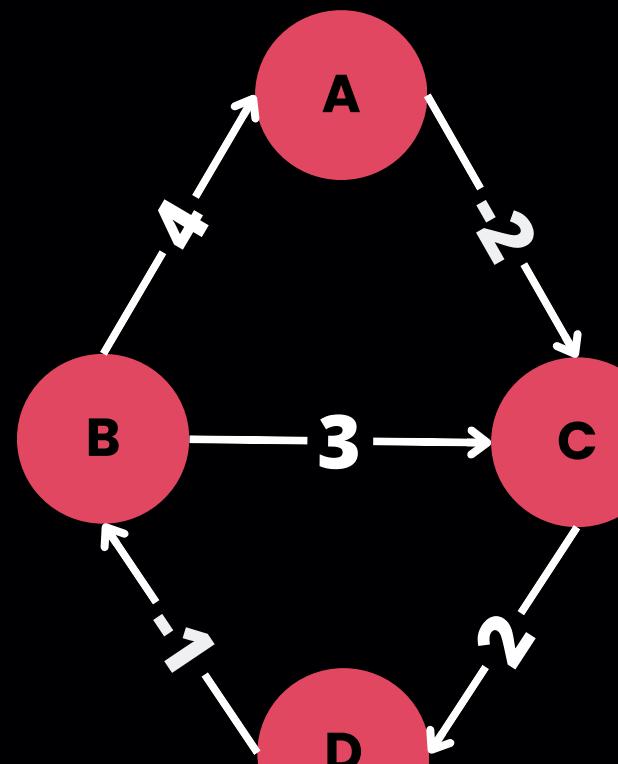
$$\text{Odl}[D][A] \rightarrow \text{Odl}[D][A] + \text{Odl}[A][C] \rightarrow \text{INF} \rightarrow -1 + 4 \rightarrow \text{INF} \rightarrow 3 \rightarrow \text{Odl}[D][A] = 3$$

ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0	INF	-2	INF
B	4	0	2	INF
C	INF	INF	0	2
D	INF	-1	INF	0



ALGORYTM FLOYDA – WARSHALLA



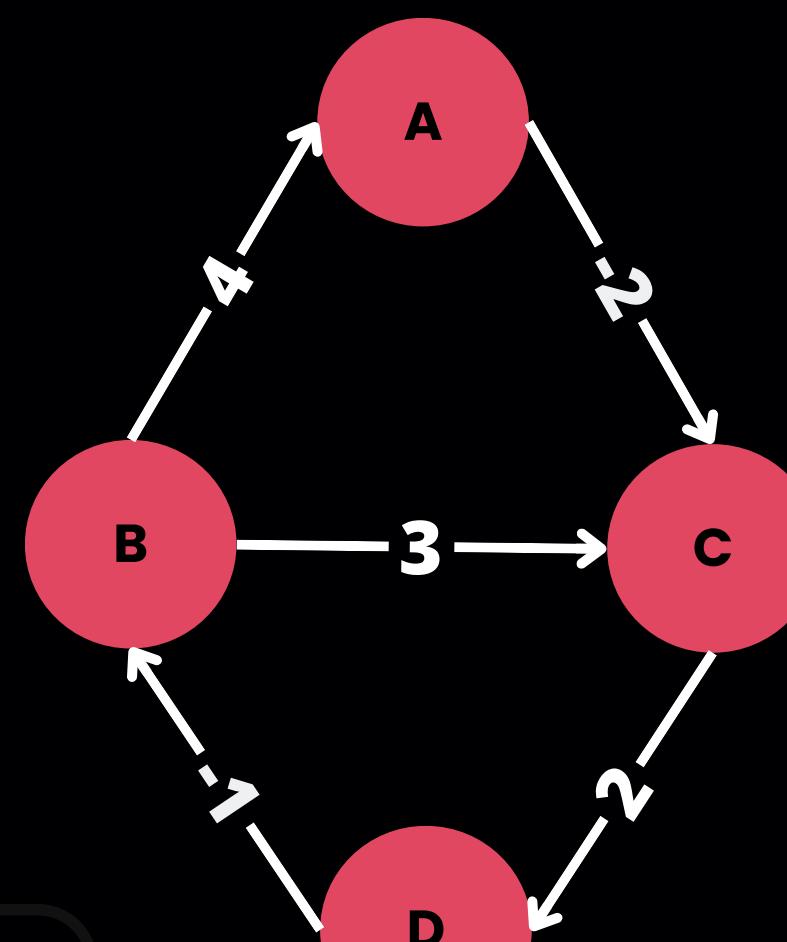
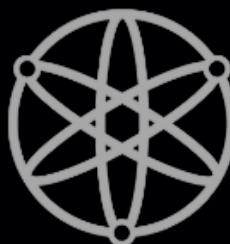
K: A B C D

I: A B C D

J: A B C D

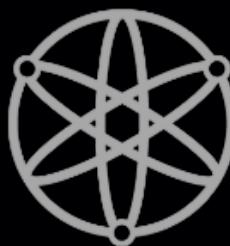
	A	B	C	D
A	0	-1	-2	0
B	4	0	2	4
C	5	INF	0	2
D	3	-1	1	0

$$\text{Odl}[c][b] \rightarrow \text{Odl}[c][d] + \text{Odl}[d][b] \rightarrow \text{INF} > 2 + -1 \rightarrow \text{INF} > 1 \rightarrow \text{Odl}[d][a] = 1$$

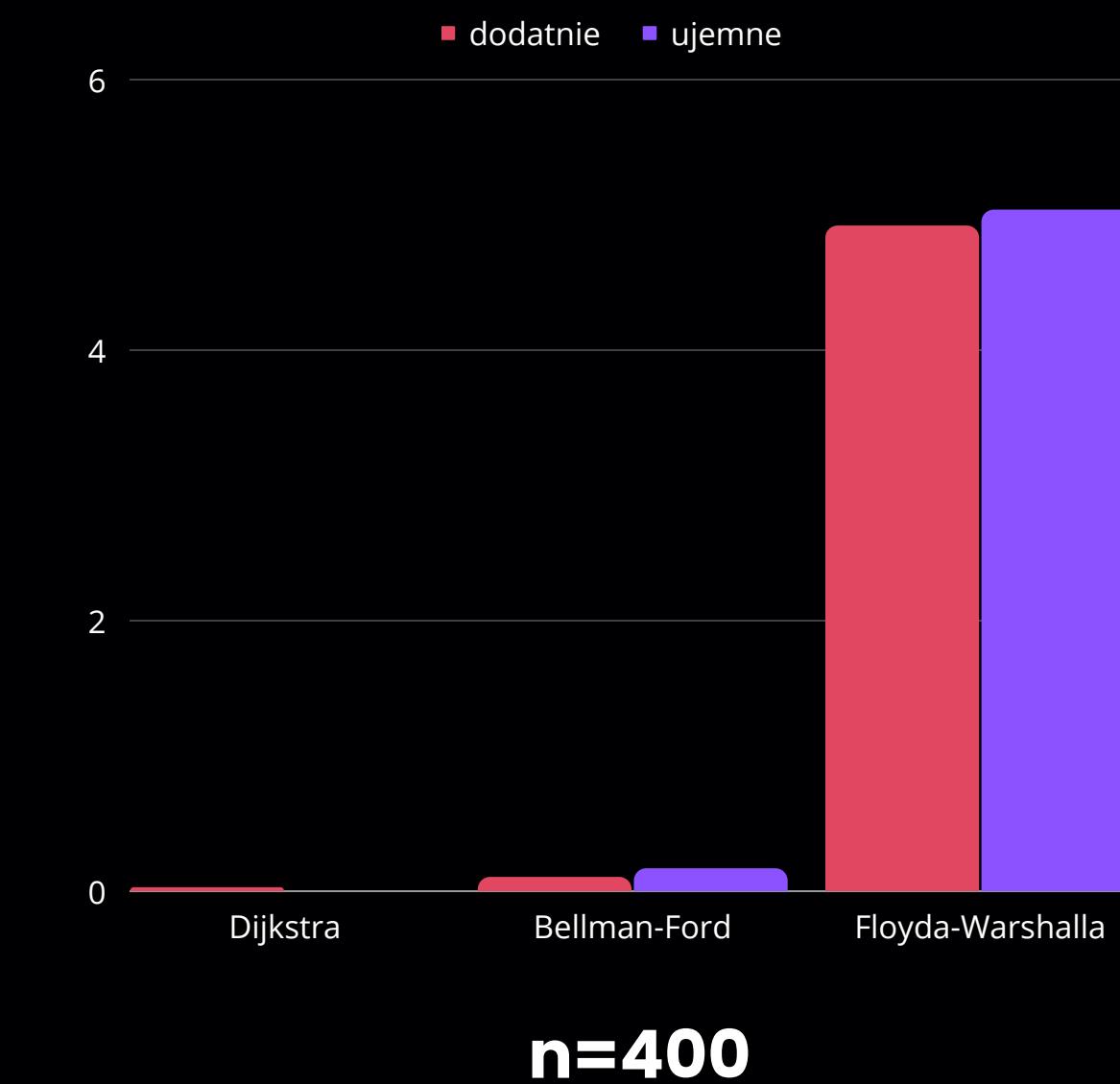
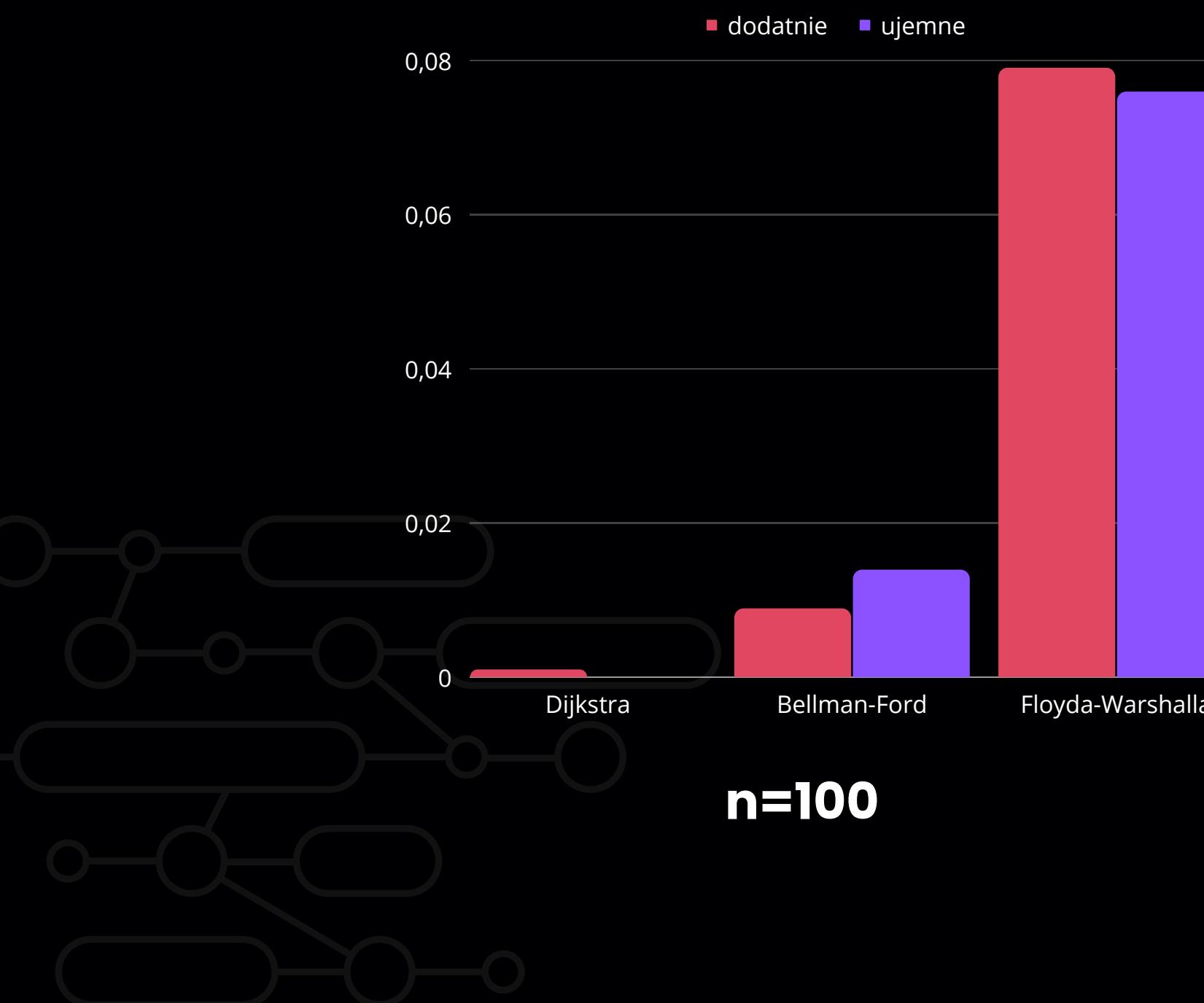


ALGORYTM FLOYDA – WARSHALLA

	A	B	C	D
A	0	-1	-2	0
B	4	0	2	4
C	5	1	0	2
D	3	-1	1	0



ZŁOŻONOŚĆ CZASOWA





WYDZIAŁ
MATEMATYKI
I FIZYKI STOSOWANEJ
POLITECHNIKI RZESZOWSKIEJ

DZIĘKUJEMY ZA
UWAGĘ