

Contents

1 Sieci, ćwiczenia 1	1
1.1 Zadanie 1	1
1.2 Zadanie 2	1
1.3 Zadanie 3	2
1.4 Zadanie 4	2
1.5 Zadanie 5	2
1.6 Zadanie 6	2
1.7 Zadanie 7	4

1 Sieci, ćwiczenia 1

tags: sieci

1.1 Zadanie 1

- 10.1.2.3/8 - jest to adres komputera. Adres sieci: 10.0.0.0/8, adres rozgłoszeniowy: 10.255.255.255/8, inny adres z sieci: 10.1.2.4/8.
- 156.17.0.0/16 - jest to adres sieci. Adres rozgłoszeniowy: 156.17.255.255/16, adres jakiegoś komputera: 156.17.21.37/16.
- 99.99.99.99/27 - prefiks to 01100011.01100011.01100011.011. Zatem jest to adres komputera. Adres sieci: 99.99.99.96/27, adres rozgłoszeniowy: 99.99.99.127/27, adres jakiegoś komputera: 99.99.99.99.100/27.
- 156.17.64.4/30 - prefiks to 157.17.64.000001. Pierwszy adres w tej sieci to właśnie 156.17.64.4/30, więc jest to adres sieci. Adres rozgłoszeniowy to 156.17.64.7/30, adres jakiegoś komputera: 156.17.64.6/30.
- 123.123.123.123/32 - prefiks jest równy temu adresowi. Jest to po prostu konkretny adres IP.

1.2 Zadanie 2

Adres pierwszej sieci: 10.10.0.0/19, Adres drugiej sieci: 10.10.32.0/19, Adres trzeciej sieci: 10.10.64.0/19, Adres czwartej sieci: 10.10.96.0/19, Adres piątej sieci: 10.10.128.0/17.

Wcześniej mieliśmy do użycia $2^{16} - 2$ adresów, teraz mamy do użycia $2^{16} - 10$ adresów, zatem o 8 mniej.

Najmniejsza podsieć jaką możemy zrobić musi mieć rozmiar co najmniej 2^{12} adresów. Załóżmy, że dałoby się zrobić podsieć, która ma 2^{11} adresów. Wtedy 4 pozostałe podsieci musiałby “sumować się” do $2^{16} - 2^{11} = 11111000\dots 0_{(2)}$, ale rozmiary to potęgi dwójki, no więc widać że się nie da (jedynek jest > 4).

1.3 Zadanie 3

- 0.0.0.0/0 ----> A
- 10.0.0.0/22 ----> B
- 10.0.1.0/24 ----> C
- 10.0.1.8/29 ----> B
- 10.0.1.16/28 ----> B

1.4 Zadanie 4

- 0.0.0.0/0 ----> A
- 10.0.0.0/8 ----> B
- 10.3.0.0/24 ----> C
- 10.3.0.32/27 ----> B
- 10.3.0.64/26 ----> B

1.5 Zadanie 5

Wystarczy je posortować po długości prefiksu (tzn. te z dłuższym prefiksem powinny mieć większy priorytet). Chcemy wybierać najbardziej “szczegółowe” reguły (czyli te z dłuższym prefiksem podsieci). W takim razie, po posortowaniu, przechodząc od lewej do prawej po tablicy, w pierwszej kolejności rozważymy najbardziej szczegółowe reguły, dopiero później przechodząc do tych mniej. Niech dla pewnego adresu ip I zbiór pasujących reguł to $\{R_1, R_2, \dots, R_k\}$, a odpowiadające im długości prefiksów to $l_1 \leq l_2 \leq \dots \leq l_k$. Wtedy, w tak ułożonej tablicy routingu jak to opisaliśmy, reguły R_1, \dots, R_k będą występowały w kolejności zadanej ciągiem l_i , zatem pierwsza reguła na jaką trafimy, to R_1 , a jest to najbardziej szczegółowa reguła dla I .

1.6 Zadanie 6

1.6.0.1 Krok 0:

	do A	do B	do C	do D	do E	do F
A	-	1				
B	1	-	1			
C		1	-		1	1
D				-	1	
E			1	1	-	1
F			1		1	-

1.6.0.2 Krok 1:

	do A	do B	do C	do D	do E	do F
A	-	1	2 (via B)			
B	1	-	1		2 (via C)	2 (via C)
C	2 (via B)	1	-	2 (via E)	1	1
D			2 (via E)	-	1	2 (via E)
E		2 (via C)	1	1	-	1
F		2 (via C)	1	2 (via E)	1	-

1.6.0.3 Krok 2:

	do A	do B	do C	do D	do E	do F
A	-	1	2 (via B)		3 (via B)	3 (via B)
B	1	-	1	3 (via C)	2 (via C)	2 (via C)
C	2 (via B)	1	-	2 (via E)	1	1
D		3 (via E)	2 (via E)	-	1	2 (via E)
E	3 (via C)	2 (via C)	1	1	-	1
F	3 (via C)	2 (via C)	1	2 (via E)	1	-

1.6.0.4 Krok 3:

	do A	do B	do C	do D	do E	do F
A	-	1	2 (via B)	4 (via B)	3 (via B)	3 (via B)
B	1	-	1	3 (via C)	2 (via C)	2 (via C)
C	2 (via B)	1	-	2 (via E)	1	1
D	4 (via E)	3 (via E)	2 (via E)	-	1	2 (via E)
E	3 (via C)	2 (via C)	1	1	-	1

	do A	do B	do C	do D	do E	do F
F	3 (via C)	2 (via C)	1	2 (via E)	1	-

1.7 Zadanie 7

Łączymy bezpośrednio ze sobą routery A oraz D.

1.7.0.1 Krok 0:

	do A	do B	do C	do D	do E	do F
A	-	1	2 (via B)	1	3 (via B)	3 (via B)
B	1	-	1	3 (via C)	2 (via C)	2 (via C)
C	2 (via B)	1	-	2 (via E)	1	1
D	1	3 (via E)	2 (via E)	-	1	2 (via E)
E	3 (via C)	2 (via C)	1	1	-	1
F	3 (via C)	2 (via C)	1	2 (via E)	1	-

1.7.0.2 Krok 1:

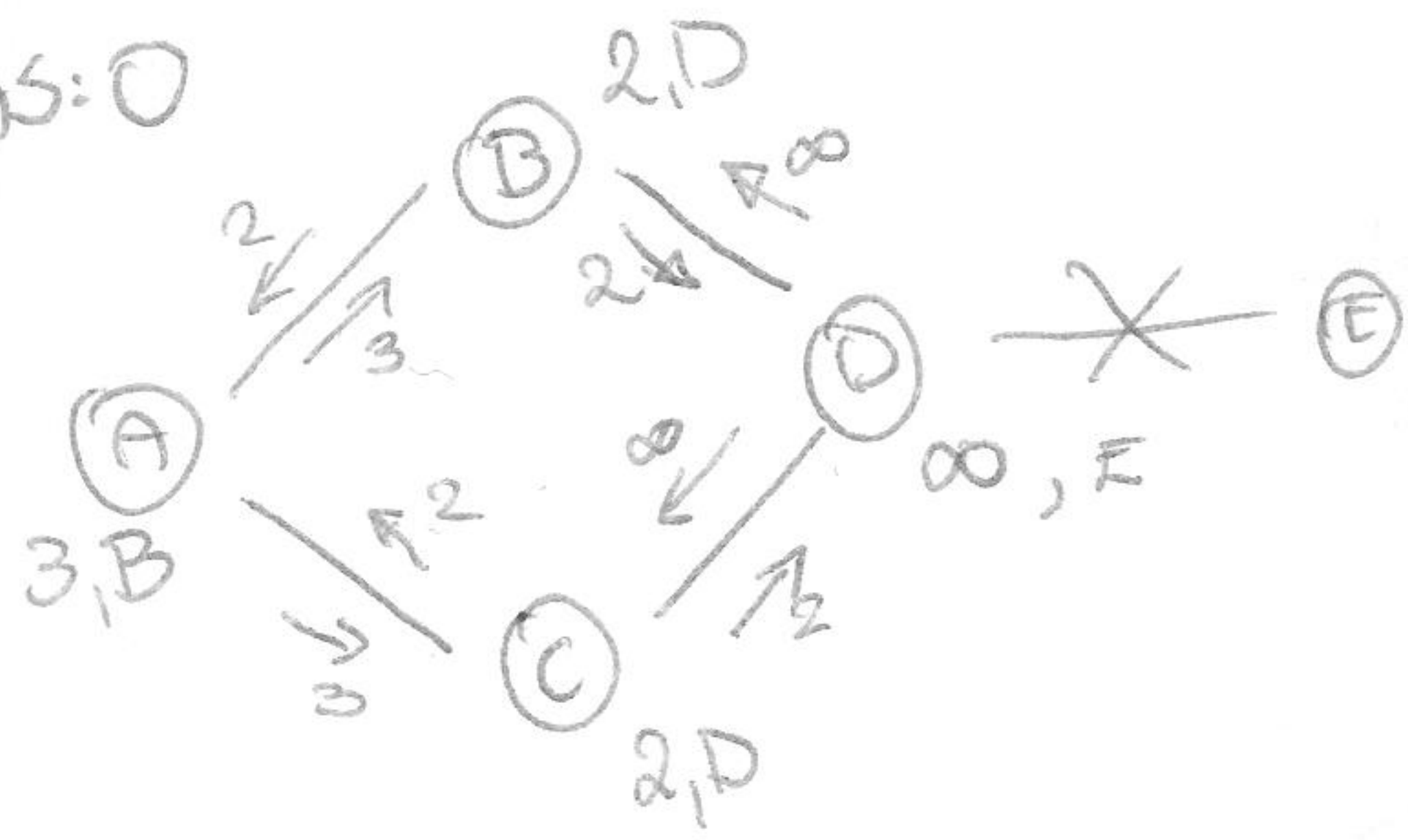
	do A	do B	do C	do D	do E	do F
A	-	1	2 (via B)	1	2 (via D)	3 (via B)
B	1	-	1	2 (via A)	2 (via C)	2 (via C)
C	2 (via B)	1	-	2 (via E)	1	1
D	1	2 (via A)	2 (via E)	-	1	2 (via E)
E	2 (via D)	2 (via C)	1	1	-	1
F	3 (via C)	2 (via C)	1	2 (via E)	1	-

Więcej się nie poprawi.

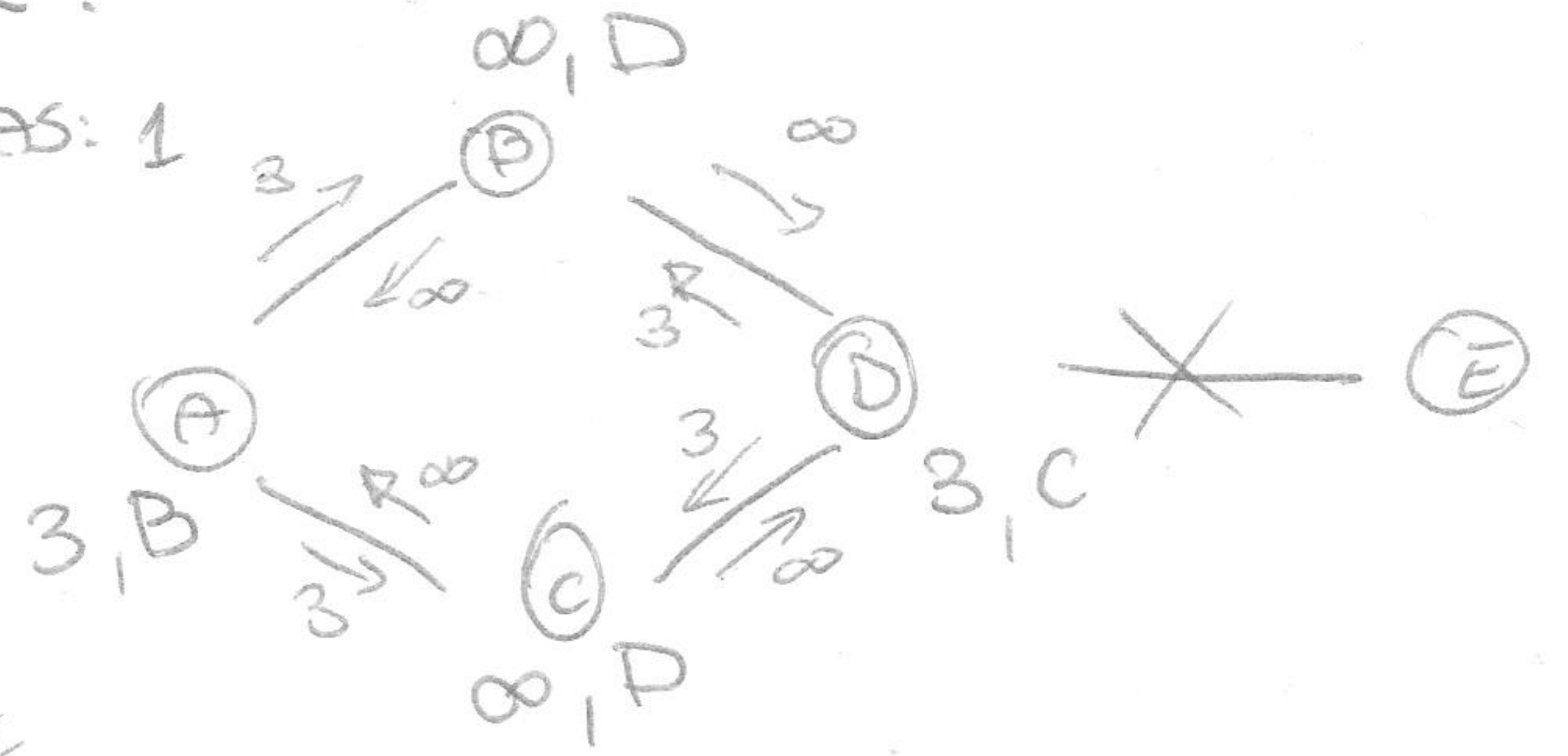
Zad. 3

Przy wierzchołkach piszemy ich odległości od E oraz 1. counter na ścieżce do E.

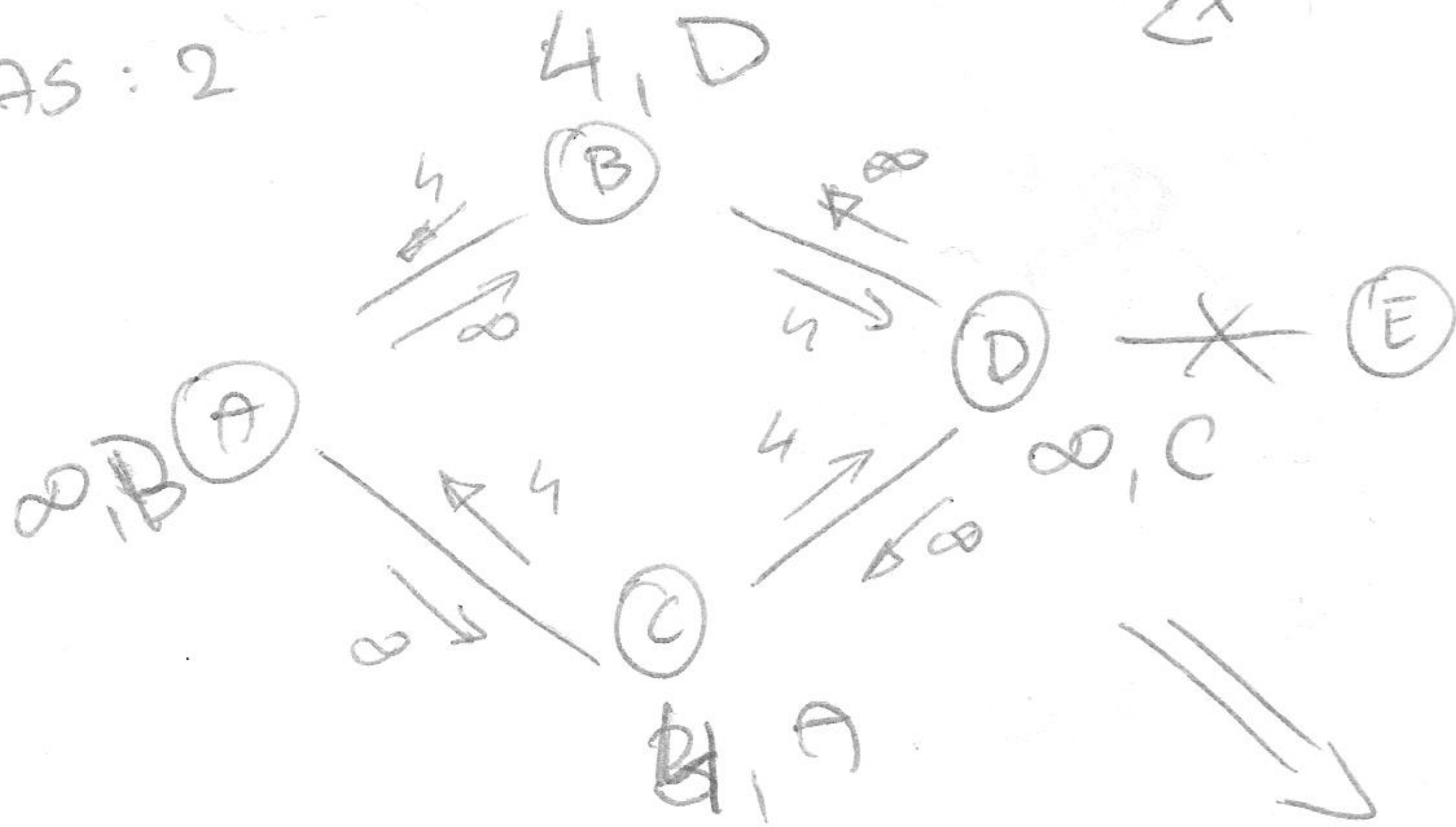
CZAS: 0



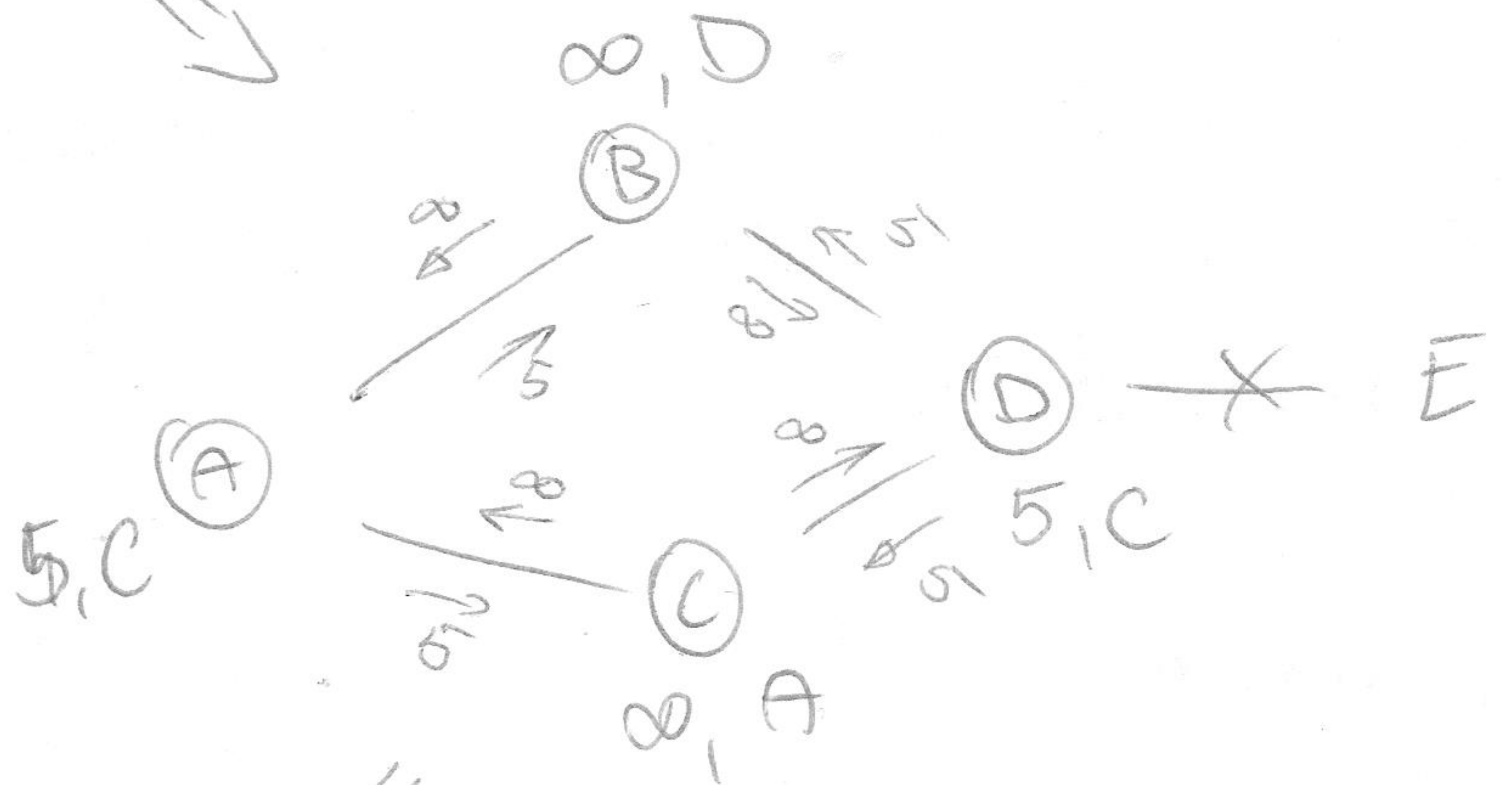
CZAS: 1



CZAS: 2



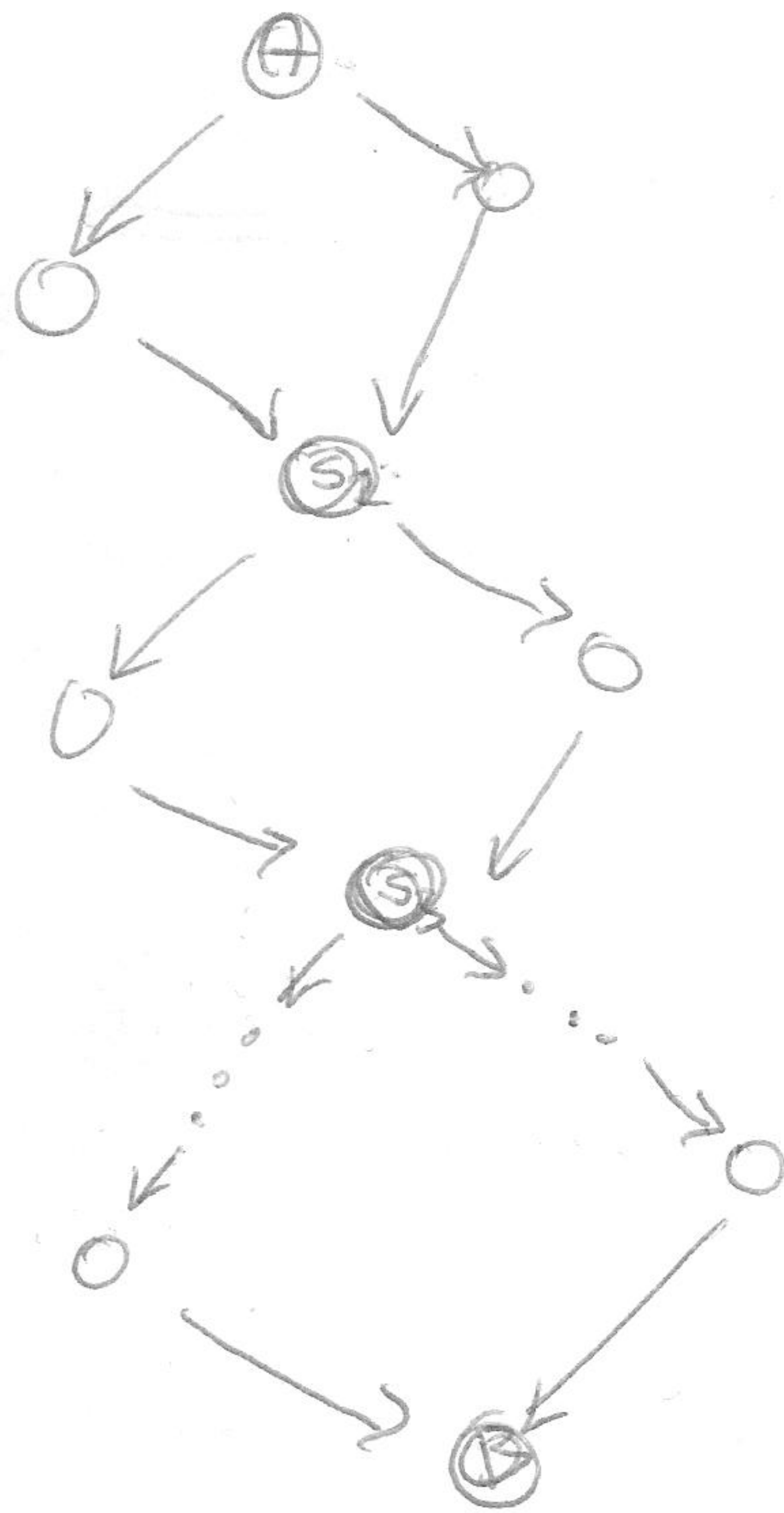
CZAS: 3



Zad. 10

BSO $n = 3k + 1$ dla pewnego $k \in \mathbb{N}$.

Wtedy sieć postaci



ma zadaną utasność. Powiedzmy, że A rozesła jakąś wiadomość. Wtedy rozsyłanie skończy się po czasie $2^{\Omega(n)}$.

D-d.

"Środkowe" routery nazwijmy $S_1 = A, S_2, \dots, S_{k+1} = B$

Udowodnimy następującą tezę przy użyciu indukcji:

~~router S_l skończy propagować swoje wiadomości~~

~~po 2^{l-1} jedn. czasu.~~

router S_l będzie musiał spropagować 2^{l-1} wiadomości.

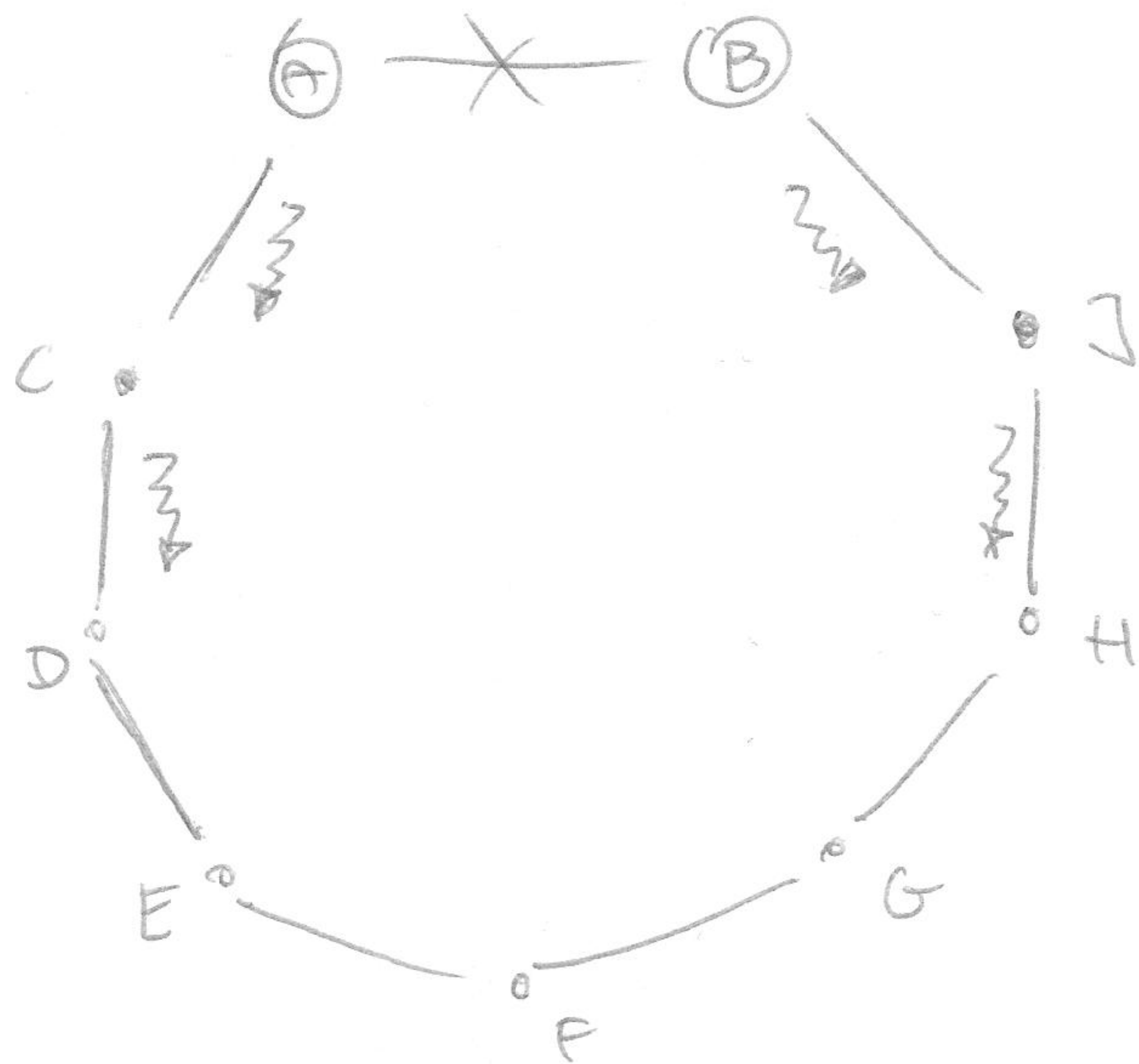
Dla $l = 1$ teza jest oczywista. ~~Dot.~~ teza dla $l-1$.

Do routera S_l , wg założenia indukcyjnego dotrze $2^{l-2} \cdot 2$ wiadomości, a ~~pro~~ zatem teza jest prawdziwa.

$$S_{k-1} = 2^{k-2} \approx 2^{\frac{n}{3}} = 2^{\Omega(n)}$$



Zad. 9



• A propaguje informację do C, B do J.

Informacja nie zdąży dojść do routera F.

Teraz, wg algorytmu najkrótszych ścieżek

router C ustali trasę do B przez D, E, ..., J,

analogicznie router J ustali trasę przez H, G, ..., C.

Teraz E ma ustaloną trasę ~~E → D~~ → D → C → A → B do

routera B. Jeśli wyślemy coś z C do B,

to dojdzie to do routera E, w którym wstąpi

ta sama trasa do routera z najbardziej

aktualnymi danymi o stanie sieci i pakiet

będzie kieszty.