

1. Dla każdego z podanych poniżej adresów IP w notacji CIDR określ, czy jest to adres sieci, adres rozgłoszeniowy czy też adres komputera. W każdym przypadku wyznacz odpowiadający mu adres sieci, rozgłoszeniowy i jakiś adres IP innego komputera w tej samej sieci.

- 10.1.2.3/8 – adres komputera
Adres sieci: 10.0.0.0
Adres rozgłoszeniowy: 10.255.255.255
Przykładowy adres: 10.1.1.1
- 156.17.0.0/16 – adres sieci
Adres sieci: 156.17.0.0
Adres rozgłoszeniowy: 156.17.255.255
Przykładowy adres: 156.17.2.137
- 99.99.99.99/27 – adres komputera
Adres sieci: 99.99.99.96
Adres rozgłoszeniowy: 99.99.99.127
Przykładowy adres: 99.99.99.100
- 156.17.64.4/30 – adres sieci
Adres sieci: 156.17.64.4
Adres rozgłoszeniowy: 156.17.64.7
Przykładowy adres: 156.17.64.5
- 123.123.123.123/32
Nie jest to adres sieci, tylko pojedynczego hosta

2. Nie zadeklarowane

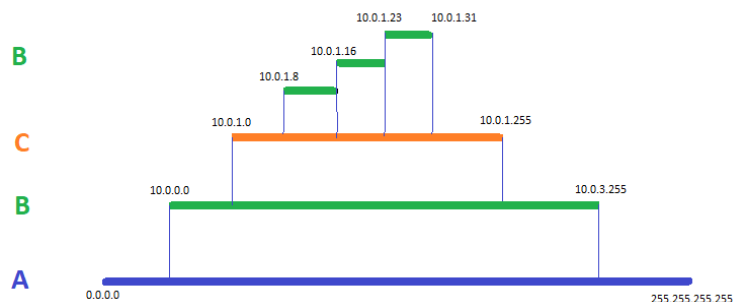
3. Tablica routingu zawiera następujące wpisy (podsieć → dokąd wysłać):

- 0.0.0.0/0 → do routera A
- 10.0.0.0/23 → do routera B
- 10.0.2.0/24 → do routera B
- 10.0.3.0/24 → do routera B
- 10.0.1.0/24 → do routera C
- 10.0.0.128/25 → do routera B
- 10.0.1.8/29 → do routera B
- 10.0.1.16/29 → do routera B
- 10.0.1.24/29 → do routera B

Napisz równoważną tablicę routingu zawierającą jak najmniej wpisów.

Zakresy routinguów:

- 0.0.0.0/0: default
- 10.0.0.0/23: 10.0.0.0 – 10.0.1.255
- 10.0.2.0/24: 10.0.2.0 – 10.0.2.255
- 10.0.3.0/24: 10.0.3.0 – 10.0.3.255
- 10.0.1.0/24: 10.0.1.0 – 10.0.1.255
- 10.0.0.128/25: 10.0.0.128 – 10.0.0.255
- 10.0.1.8/29: 10.0.1.8 – 10.0.1.15
- 10.0.1.16/29: 10.0.1.16 – 10.0.1.23
- 10.0.1.24/29: 10.0.1.24 – 10.0.1.31



Zmniejszona tablica routingu:

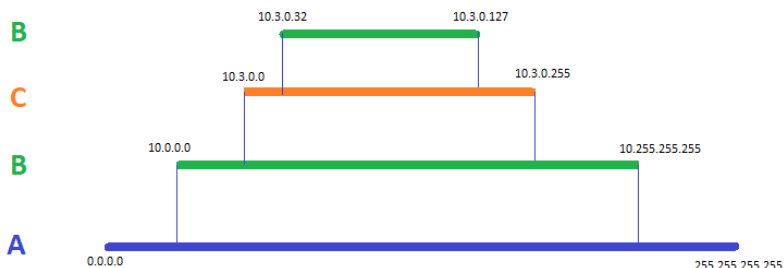
- 0.0.0.0/0 → do A
- 10.0.0.0/22 → do B
- 10.0.1.0/24 → do C
- 10.0.1.8/29 → do B
- 10.0.1.16/28 → do B

4. Wykonaj powyższe zadanie dla tablicy

- 0.0.0.0/0 → do routera A
- 10.0.0.0/8 → do routera B
- 10.3.0.0/24 → do routera C
- 10.3.0.32/27 → do routera B
- 10.3.0.64/27 → do routera B
- 10.3.0.96/27 → do routera B

Zakresy:

- 0.0.0.0/0: default
- 10.0.0.0/8: 10.0.0.0 – 10.255.255.255
- 10.3.0.0/24: 10.3.0.0 – 10.3.0.255
- 10.3.0.32/27: 10.3.0.32 – 10.3.0.63
- 10.3.0.64/27: 10.3.0.64 – 10.3.0.95
- 10.3.0.96/27: 10.3.0.96 – 10.3.0.127



Patrząc na rysunek można zauważyć, że górna sieć B nadpisuje C, zatem można B złączyć w całość, a C zredukować do 2 mniejszych zakresów.

Zmniejszona tablica routingu:

- 0.0.0.0/0 → do A
- 10.0.0.0/8 → do B
- 10.3.0.0/27 → do C
- 10.3.0.128/25 → do C

5. Jak uporządkować wpisy w tablicy routingu, żeby zasada najlepszego dopasowania odpowiadała wybo- rowi „pierwszy pasujący” (tj. przeglądaniu tablicy od początku do końca aż do momentu napotkania dowolnej pasującej reguły)? Odpowiedź uzasadnij formalnie.

Wpisy w tablicy będziemy uporządkowywać od najdłuższych pasujących prefiksów do najkrótszych. Jeżeli długość prefiksu jest taka sama, to nie sortujemy.

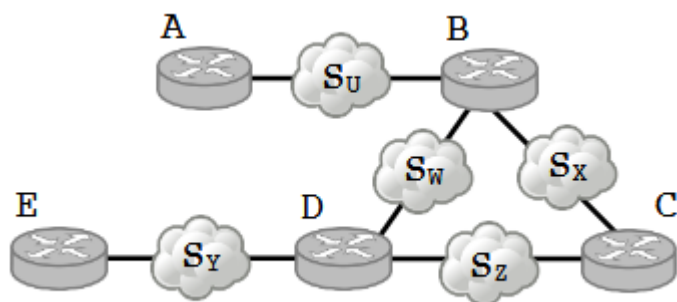
Dowód:

Założmy, że wpisy w tablicy routingu są uporządkowane od najdłuższych pasujących prefiksów do najkrótszych.

Udowodnimy, że wybór pierwszego pasującego wpisu prowadzi do najlepszego dopasowania:

Weźmy dowolny adres IP, oznaczmy go jako x. Niech y będzie pierwszym wpisem w tablicy routingu, do którego nie pasuje x. Przez n oznaczmy długość prefiksu y, a skoro x pasuje do y, to pierwsze n bitów x oraz y są takie same. Teraz weźmy dowolny wpis z będący w tablicy routingu za y. Oznacza to, że z ma mniejszy prefiks od y, a więc długość m prefiksu z spełnia nierówność $m \leq n$. Oznacza to, że dopasowane zostanie nie więcej bitów x niż n, czyli y nie jest gorszym dopasowaniem niż z, więc wybór "pierwszy pasujący" odpowiada zasadzie najlepszego dopasowania.

6. W podanej niżej sieci tablice routingu budowane są za pomocą algorytmu wektora odległości. Pokaż (krok po kroku), jak będzie się to odbywać. W ilu krokach zostanie osiągnięty stan stabilny?



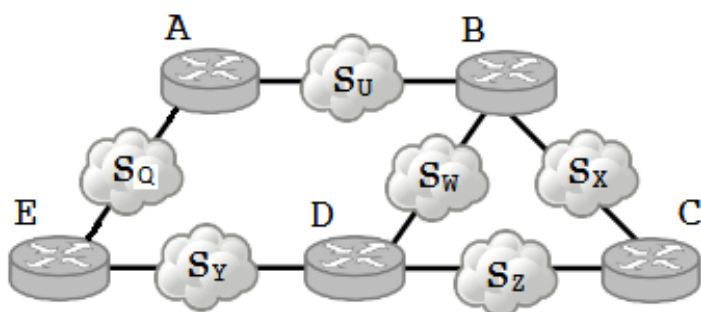
Krok 1.	A	B	C	D	E
do Su	1	1			
do Sw		1		1	
do Sx		1	1		
do Sz			1	1	
do Sy				1	1

Krok 2.	A	B	C	D	E
do Su	1	1	via B(2)	via B(2)	
do Sw	via B(2)	1	via B(2)	1	via D(2)
do Sx	via B(2)	1	1	via B(2)	
do Sz		via C(2)	1	1	via D(2)
do Sy		via D(2)	via D(2)	1	1

Krok 3.	A	B	C	D	E
do Su	1	1	via B(2)	via B(2)	via D(3)
do Sw	via B(2)	1	via B(2)	1	via D(2)
do Sx	via B(2)	1	1	via B(2)	via D(3)
do Sz	via B(3)	via C(2)	1	1	via D(2)
do Sy	via B(3)	via D(2)	via D(2)	1	1

Jak widać w 3. kroku osiągniemy stan stabilny - wszystkie routery w sieci będą znały drogę do pozostałych, a wyznaczone ścieżki są najkrótsze.

7. Załóżmy, że w powyższej sieci tablice routingu zostały już zbudowane. Co będzie się działo (krok po kroku), jeśli zostanie dodana sieć Sq łącząca routery A i E?

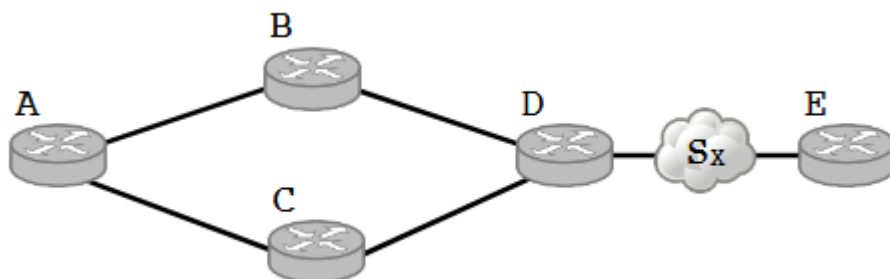


Krok 1.	A	B	C	D	E
do Su	1	1	via B(2)	via B(2)	via D(3)
do Sw	via B(2)	1	via B(2)	1	via D(2)
do Sx	via B(2)	1	1	via B(2)	via D(3)
do Sz	via B(3)	via C(2)	1	1	via D(2)
do Sy	via B(3)	via D(2)	via D(2)	1	1
do Sq	1				1

Krok 2.	A	B	C	D	E
do Su	1	1	via B(2)	via B(2)	via A(2)
do Sw	via B(2)	1	via B(2)	1	via D(2)
do Sx	via B(2)	1	1	via B(2)	via D(3)
do Sz	via B(3)	via C(2)	1	1	via D(2)
do Sy	via D(2)	via D(2)	via D(2)	1	1
do Sq	1	via A(2)	via B(3)	via E(2)	1

Osiągnięty stan stabilny, wszystkie ścieżki są najkrótsze.

8. W przedstawionej poniżej sieci uszkodzeniu ulega połączenie między routerami D i E. Załóżmy, że w sieci działa algorytm wektora odległości wykorzystujący technikę zatruwania ścieżki zwrotnej (poison reverse). Pokaż — opisując krok po kroku jakie komunikaty są przesyłane między routerami — że może powstać cykl w routingu.



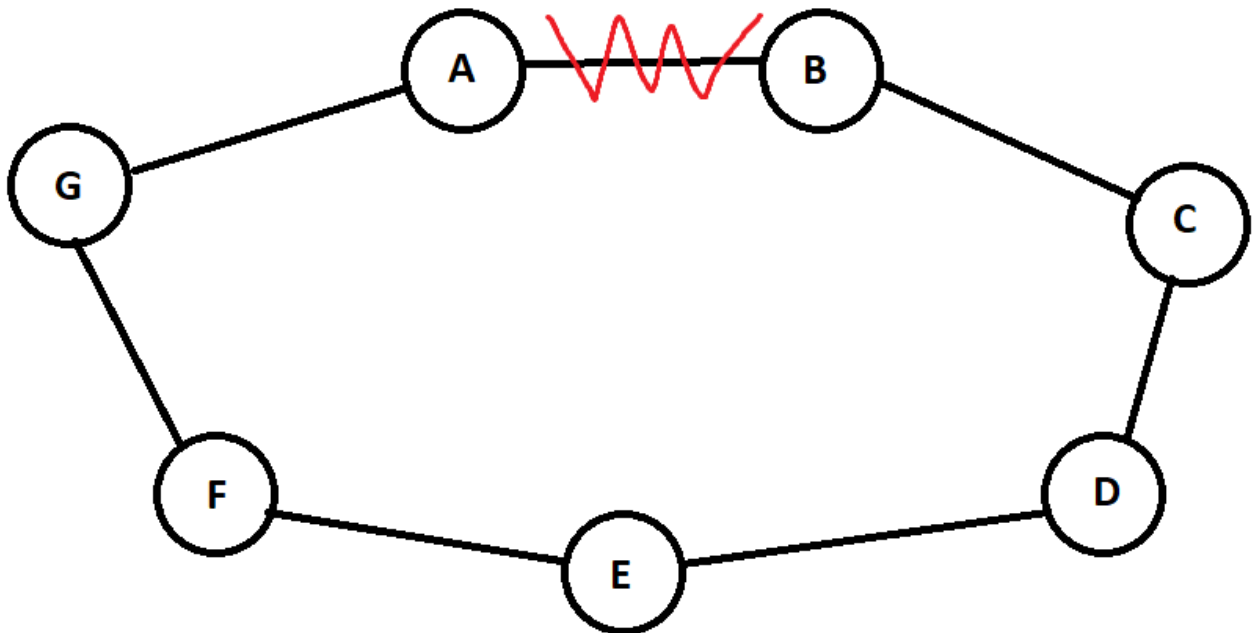
Przeanalizujemy zmiany tabeli routing od momentu rozłączenia:

krok łączenia do E	A	B	C	D
0	via C(3)	via D(3)	via D(2)	inf
1	via C(3)	inf	via D(2)	inf
2	via C(3)	via A(4)	inf	inf
3	inf	via A(4)	inf	via B(5)
4	inf	inf	via D(6)	via B(5)

itd.... widać, że powstaje zapętlenie.

9. Pokaż, że przy wykorzystaniu algorytmu stanu łączy też może powstać cykl w routingu. W tym celu skonstruuj sieć z dwoma wyróżnionymi, sąsiadującymi ze sobą routerami A i B. Załóż, że wszystkie routery znają graf całej sieci. W pewnym momencie łącze między A i B ulega awarii, o czym A i B od razu się dowiadują. Zalewają one sieć odpowiednią aktualizacją. Pokaż, że w okresie propagowania tej aktualizacji (kiedy dotarła ona już do części routerów a do części nie) może powstać cykl w routingu.

Weźmy przykładowy graf sieci:

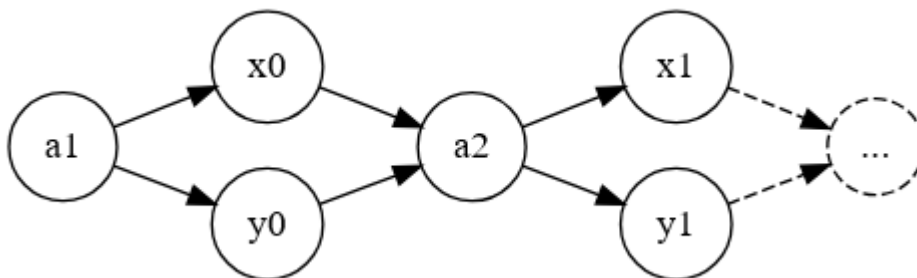


W momencie uszkodzenia łącza A-B: A i B wysyłają o tym informację do reszty routerów w sieci. Może tak się stać, że router C dostanie te informacje w tym samym momencie co informacje z routera D, że posiada on połączenie do A (które idzie starą drogą przez C i B).

W takiej sytuacji router C będzie próbował wysłać pakiety do routera A przez router D, a router D będzie je wysyłał z powrotem do C i tak w kółko, przez co powstanie małe zapętlenie do momentu aż z drugiej strony dotrze informacja o uszkodzeniu łącza A-B i router D zacznie poprawnie przysyłać pakiety „w drugą stronę”.

10. Załóżmy, że sieć składa się z łączy jednokierunkowych (tj. łącza w sieci tworzą graf skierowany) i nie zawiera cykli. Rozważmy niekontrolowany algorytm „zalewający” sieć jakimś komunikatem: komunikat zostaje wysłany początkowo przez pewien router; każdy router, który dostanie dany komunikat przesyła go dalej wszystkimi wychodzącymi z niego krawędziami. Pokaż, że istnieją takie sieci z n routerami, w których przesyłanie informacji zakończy się po czasie $2\Omega(n)$. Zakładamy, że przez jedno łącze można przesłać tylko jeden komunikat naraz, a przesłanie go trwa jednostkę czasu.

Rozważmy taki układ:



Router a_1 będzie wysyłał jeden komunikat do 2 routerów x_0, y_0 , przez co router a_2 będzie miał do wysłania dwa osobne komunikaty w dwóch jednostkach czasu do routerów x_1, y_1 itd. ciągle podwajając ilość komunikatów.

Wiedząc, że mamy n routerów, czyli $n/3$ podwojeń to da nam $2 * \sum_{i=0}^{n/3} 2^i \approx 2^{1+\frac{n}{3}} - 1$ jednostek czasu.

A $1 + n/3 \in \Omega(n)$, więc taki układ spełnia wymogi zadania.