

Bartosz Kruszewski 337568

Sieci Komputerowe Ćwiczenia 1

zadanie 1

Adresy sieci są pierwsze w danej sieci.

Adresy rozgłoszeniowe są ostatnie w danej sieci.

Adresy komputerów są pomiędzy.

Adres CIDR	Adres sieci	Adres sieci	Adres rozgłoszeniowy	Przykładowy adres komputera
10.1.2.3/8	10.0.0.0	Adres komputera	10.255.255.255	10.0.0.1
156.17.0.0/16	156.17.0.0	Adres sieci	156.17.255.255	156.17.0.1
99.99.99.99/27	99.99.99.96	Adres komputera	99.99.99.127	99.99.99.97
156.17.64.4/30	156.17.64.4	Adres sieci	156.17.64.7	156.17.64.5
123.123.123.123/32	123.123.123.123	Oba na raz	123.123.123.123	Brak dostępnych hostów

zadanie 2

Rozmiar podsieci może być tylko potęgą 2.

Sieć **10.10.0.0/16** ma **2^{16} adresów** (w tym dwa nieużyteczne).

Chcemy pokryć wszystkie adresy, więc: - najmniejsza wielkość sieci musi wystąpić dwa razy, ponieważ muszą się sumować do wyższej potęgi 2.

- jeżeli chcemy żeby sieć zajmowała najmniej to pozostałe sieci muszą zajmować jak najwięcej.

$$2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{12} = 2^{16}.$$

Wyznaczmy podsieci takich wielkości: - 10.10.128.0/17 - 10.10.64.0/18 - 10.10.32.0/19 - 10.10.16.0/20 - 10.10.0.0/20

Minimalny rozmiar podsieci to $2^{12} - 2 = 4096 - 2 = 4094$
(Odejmujemy dwa bo to adres sieci i rozgłoszeniowy)

zadanie 3

Stan początkowy:

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

Zmiany:

10.0.0.128/25 → do routera B
stanowi podsiec
10.0.0.0/23 → do routera B
wiec mozemy go usunac

10.0.2.0/24 → do routera B
10.0.3.0/24 → do routera B
mozemy skleic w
10.0.2.0/23 → do routera B

10.0.2.0/23 → do routera B
mozemy skleic z
10.0.0.0/23 → do routera B
tworzac
10.0.0.0/22 → do routera B

10.0.1.16/29 → do routera B
10.0.1.24/29 → do routera B
mozemy skleic w
10.0.1.16/28 → do routera B

Ostatecznie otrzymamy:

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

zadanie 4

Stan początkowy:

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

Zmiany:

10.3.0.64/27 → do routera B
10.3.0.96/27 → do routera B
sklejamy w
10.3.0.64/26 → do routera B

10.3.0.32/27 → do routera B
10.3.0.64/26 → do routera B
zawierają się w
10.0.0.0/8 → do routera B
ale również w
10.3.0.0/24 → do routera C

możemy więc usunąć je albo zmodyfikować
10.3.0.0/24 → do routera C
na dwa wpisy
10.3.0.0/27 → do routera C
10.3.0.128/25 → do routera C

Ostatecznie otrzymamy:

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

zadanie 5

Najlepsze dopasowanie w tablicy routingu to takie które ma najwięcej pasujących pierwszych bitów.

Stąd żeby uzyskać efekt “pierwszy dopasowany - najlepszy dopasowany” najlepiej posortować wpisy po długości adresu (malejąco względem wartości *CIDR*).

Dowód formalny:

Założenia: Tablica jest posortowana malejąco względem długości bitów

Teza: Pierwsze dopasowanie będzie najlepszym dopasowaniem

Weźmy adres i znajdziemy jego pierwsze dopasowanie (Zakładamy że jakieś dopasowanie istnieje, inaczej i tak nie ma z czym porównać). Ponieważ adres został dopasowany to jego x pierwszych bitów jest dopasowane. Weźmy dowolne inne dopasowanie, wtedy y bitów zostało dopasowane. Z założenia, że tablica jest posortowana malejąco wiemy że dowolny inny adres był tak samo długi lub krótszy. Stąd $x \geq y$, więc z definicji pierwsze dopasowanie jest tak samo dobre lub lepsze niż dowolne inne, więc jest najlepsze.

zadanie 6

Algorytm polega na przechowywanie przez routery wektora odleglosci do kazdego innego routera oraz ich okresow rozsyłaniu miedzy soba w celu zaktualizowania tych wektorow.

Początkowo wektor odleglosci zawiera tylko informacje o odleglosci do swoich bezpośrednich sąsiadow. Następnie dodaje odleglosci z wektora otrzymanego od sąsiadow, bierze minimum wszystkich wynikow i aktualizuje swoje odleglosci.

Stan początkowy:

	A	B	C	D	E
A	-	1			
B	1	-	1	1	
C		1	-	1	
D		1	1	-	1
E				1	-
Su	1	1			
Sw		1		1	
Sx		1	1	1	
Sy				1	1
Sz			1	1	

1. iteracja:

	A	B	C	D	E
A	-	1	2 (via B)	2 (via B)	
B	1	-	1	1	2 (via D)
C	2 (via B)	1	-	1	2 (via D)
D	2 (via B)	1	1	-	1
E		2 (via D)	2 (via D)	1	-
Su	1	1	2 (via B)	2 (via B)	
Sw	2 (via B)	1	2 (via B)	1	2 (via D)
Sx	2 (via B)	1	1	1	
Sy		2 (via D)	2 (via D)	1	1
Sz		2 (via C)	1	1	2 (via D)

2. iteracja:

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

zadanie 7

Stan początkowy (jak w poprzednim zadaniu ale A i E wiedza o sobie):

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

1. iteracja:

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

2. iteracja:

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

zadanie 8

Tabela routingu przed awarią (koszt do E)

Router	Przez	Koszt
A	B	3
A	C	3
B	D	2
C	D	2
D	E	1

Awaria połączenia D - E

Router	Przez	Koszt
D	-	∞

D oznacza, że E jest nieosiągalne i ogłasza to sąsiadom (B, C).

B i C otrzymują aktualizację od D

Router	Przez	Koszt
B	D	∞
C	D	∞

Oba routery wiedzą, że D już nie ma połączenia z E.

B i C szukają alternatywnej drogi

Router	Przez	Koszt
B	C	3
C	B	3

Każdy z nich zakłada, że drugi nadal ma działającą trasę do E i zwiększa koszt.

Powstanie pętli

Router	Przez	Koszt
B	C	4
C	B	4

Koszt wzrasta, ale pakiety krążą między B i C w pętli.

Kontynuacja błędnych aktualizacji

Router	Przez	Koszt
B	C	5
C	B	5

Każda aktualizacja zwiększa koszt, ale sieć nie może dotrzeć do E.

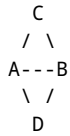
“Liczenie do nieskończoności”

Router	Przez	Koszt
B	C	∞
C	B	∞

Po wielu iteracjach koszt rośnie do nieskończoności, aż algorytm się ustabilizuje.

zadanie 9

Plan sieci



Założmy, że **łącze między A i B ulega awarii**.

1. **A i B natychmiast się o tym dowiadują i rozgłaszają aktualizację.**
2. **Informacja o awarii propaguje się stopniowo** do pozostałych routerów C i D.

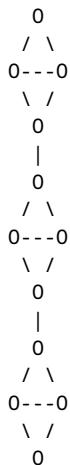
Założmy, że **router C już dostał informację o awarii**, ale router D jeszcze nie.

3. **C widzi, że łącze A-B jest zerwane**, więc jeśli ma przesłać pakiet do D, wybierze ścieżkę **C → A → D**.
4. **D jeszcze nie dostał informacji o awarii**, więc wciąż widzi starą topologię, gdzie łącze A-B istnieje. Jeśli dostanie pakiet przeznaczony dla A, może go wysłać **do B, ponieważ myśli, że łącze A-B działa**. Następnie **B, który już wie o awarii, przekieruje pakiet z powrotem do D przez inne dostępne ścieżki**.

W ten sposób **pakiet może krążyć w nieskończonej pętli**, zanim wszystkie routery zsynchronizują swoją wiedzę o sieci.

zadanie 10

Plan sieci



Zakładając że na wejściu takiego romba mamy k kopii komunikatów rozsyłamy je do dwóch wewnętrznych routerów romba a z nich do wyjścia. Wyjście ma teraz $2k$ kopii komunikatów ale tylko jedna linie do przesyłania więc przesłanie zajmie $2k$ czasu.

Zauważmy że romb składa się z czterech routerów, więc mamy $n/4$ rombów, więc do ostatniego rombu trafi $2^{n/4} - 1$ kopii, co spełnia $2^{\Omega(n)}$