Ćwiczenia z Sieci komputerowych
Lista 1

Jakub Grobelny (300481)

Zadanie	1	2	3	4	5	6	7	8	9	10
Rozwiązane	✓	✓	✓	✓	✓	✓	✓			

Zadanie 1.

- `10.1.2.3/8`
  Prefiks ma 8 bitów, czyli adres sieci to pierwszy bajt i same zera (`10.0.0.0/8`).
  Adres rozgłoszeniowy to `10.255.255.255/8`.
  `10.1.2.3/8` **jest adresem komputera**.
  Przykładowy adres innego komputera: `10.1.2.4/8`.
- `156.17.0.0/16`
  Prefiks ma 16 bitów, czyli adres **jest adresem sieci** `156.17.0.0/16` bo od szesnastego bitu są same zera.
  Adres rozgłoszeniowy: `156.17.255.255/16`
  Przykładowy adres komputera: `156.17.42.42/16`.
- `99.99.99.99/27`
  W postaci binarnej (podkreślony prefiks z 27 bitów):
  <u>`01100011 01100011 01100011 011`</u>`00011`
  Widać zatem, że **jest to adres komputera** (sufiks nie składa się z samych zer ani z samych jedynek).
  Adres sieci to `99.99.99.96/27` (prefiks + 5 zer na końcu).
  Adres rozgłoszeniowy to `99.99.99.127/27` (prefiks + 5 jedynek na końcu).
  Przykładowy adres innego komputera: `99.99.99.100/27`
- `156.17.64.4/30`
  Binarnie: <u>`10011100 00010001 01000000 000001`</u>`00`
  Widać, że adres jest pierwszym o tym prefiksie, więc **jest to adres sieci**.
  Adres rozgłoszeniowy: `156.17.64.7/30`.
  Przykładowy adres komputera: `156.17.64.6/30`.
- `123.123.123.123/32`
  Prefiks to wszystkie cztery bajty, więc **jest to jeden konkretny adres komputera** a nie zakres adresów (czyli adres sieci = adres rozgłoszeniowy = adres komputera).

Zadanie 2. `10.10.0.0/16` = 00001010 00001010 00000000 00000000.

Żeby uzyskać 5 podsieci, musielibyśmy poświęcić dodatkowe 3 bity na maskę podsieci. Przy użyciu trzech bitów można zakodować aż 8 różnych wartości, więc 3 z nich by się zmarnowały.

Rozwiązanie: dzielimy sieć na 4 podsieci, a następnie jedną z tych podsieci dzielimy jeszcze na dwie, co da nam łącznie pięć sieci.

Dostajemy więc następujące podsieci:

Zakres adresów	Trzeci bajt adresu
`10.10.0.0/18`	00000000
`10.10.64.0/18`	01000000
`10.10.128.0/18`	10000000
`10.10.192.0/19`	11000000
`10.10.224.0/19`	11100000

Są one oczywiście rozłączne, a każdy adres z zakresu `10.10.0.0/16` należy do którejś z nich.

Sieć `10.10.0.0/16` miała 65536 możliwych adresów, z czego dwa z nich to adres sieci i adres rozgłoszeniowy (co daje 65534 adresy dla komputerów).

Po podzieleniu na pięć podsieci również mamy 65536 adresów, ale każda podsieć ma teraz swój własny adres sieci i adres rozgłoszeniowy, co daje nam $65536 - 5 \cdot 2 = 65526$ adresów. **Straciliśmy więc osiem możliwych adresów możliwych do użycia przy adresowaniu komputerów**.

Pytanie: jaki jest minimalny rozmiar podsieci, który możesz uzyskać w ten sposób?

Odpowiedź: minimalną podsieć możemy uzyskać poprzez podział polegający na dzieleniu najmniejszej z uzyskanych podsieci na dwie kolejne. Taki podział wyglądałby następująco:

Zakres adresów	Trzeci bajt adresu
`10.10.0.0/17`	00000000
`10.10.128.0/18`	10000000
`10.10.192.0/19`	11000000
`10.10.224.0/20`	11100000
`10.10.240.0/20`	11110000

Jak widać w tabeli, najmniejsze dwie podsieci mają jedynie $2^{12} = 4096$ możliwych adresów, czyli tylko 4094 adresów możliwych do użycia przy adresowaniu komputerów.

Zadanie 3. Oryginalna tablica routingu:

	Adresy	Dokąd wysłać	Zakres adresów
1	`0.0.0.0/0`	router A	`0.0.0.0 – 255.255.255.255`
2	`10.0.0.0/23`	router B	`10.0.0.0 – 10.0.1.255`
3	`10.0.2.0/24`	router B	`10.0.2.0 – 10.0.2.255`
4	`10.0.3.0/24`	router B	`10.0.3.0 – 10.0.3.255`
5	`10.0.1.0/24`	router C	`10.0.1.0 – 10.0.1.255`
6	`10.0.0.128/25`	router B	`10.0.0.128 – 10.0.0.255`
7	`10.0.1.8/29`	router B	`10.0.1.8 – 10.0.1.15`
8	`10.0.1.16/29`	router B	`10.0.1.16 – 10.0.1.23`
9	`10.0.1.24/29`	router B	`10.0.1.24 – 10.0.1.31`

Wpisy 8 i 9 różnią się jedynie na ostatnim bicie prefiksu a ich prefiksy są takiej samej długości. Możemy więc złączyć je w jeden zakres `10.0.1.16/28` z prefiksem krótszym o 1 bit.

Tę samą metodę możemy zastosować do wpisów 3 i 4. Po złączeniu utworzą one zakres `10.0.2.0/23`. Wówczas jednak możemy powtórzyć to samo dla nowego zakresu oraz 2. Różnią się jedynie dwudziestym trzecim (ostatnim) bitem prefiksu, więc można utworzyć większy zakres `10.0.0.0/22`.

Widać również, że wpis 6, czyli `10.0.0.128/25`, jest podzbiorem nowo utworzonego `10.0.0.0/22`. Jednocześnie `10.0.0.128/25` jest rozłączny z `10.0.1.0/24`, więc możemy go usunąć.

Po zmniejszeniu:

	Adresy	Dokąd wysłać	Zakres adresów
1	`0.0.0.0/0`	router A	`0.0.0.0 – 255.255.255.255`
2*	`10.0.0.0/22`	router B	`10.0.0.0 – 10.0.3.255`
5	`10.0.1.0/24`	router C	`10.0.1.0 – 10.0.1.255`
8*	`10.0.1.16/28`	router B	`10.0.1.16 – 10.0.1.31`
7	`10.0.1.8/29`	router B	`10.0.1.8 – 10.0.1.15`

Zadanie 4. Oryginalna tablica routingu:

	Adresy	Dokąd wysłać	Zakres adresów
1	`0.0.0.0/0`	router A	`0.0.0.0 – 255.255.255.255`
2	`10.0.0.0/8`	router B	`10.0.0.0 – 10.255.255.255`
3	`10.3.0.0/24`	router C	`10.3.0.0 – 10.3.0.255`
4	`10.3.0.32/27`	router B	`10.3.0.32 – 10.3.0.63`
5	`10.3.0.64/27`	router B	`10.3.0.64 – 10.3.0.95`
6	`10.3.0.96/27`	router B	`10.3.0.96 – 10.3.0.127`

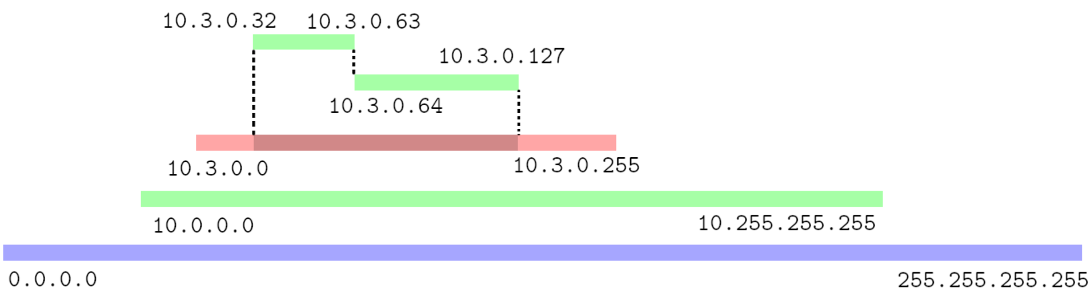
Analogicznie do poprzedniego zadania, można zredukować 5 i 6 do jednego wpisu `10.3.0.64/26` poprzez skrócenie prefiksu o bit.

Zmniejszona tabela:

	Adresy	Dokąd wysłać	Zakres adresów
1	`0.0.0.0/0`	router A	`0.0.0.0` – `255.255.255.255`
2	`10.0.0.0/8`	router B	`10.0.0.0` – `10.255.255.255`
3	`10.3.0.0/24`	router C	`10.3.0.0` – `10.3.0.255`
5*	`10.3.0.64/26`	router B	`10.3.0.64` – `10.3.0.127`
4	`10.3.0.32/27`	router B	`10.3.0.32` – `10.3.0.63`

Można jednak zauważyć, że gdybyśmy byli w stanie „wyciąć dziurę” we wpisie 3, która miałaby zakres adresów od `10.3.0.32` do `10.3.0.128`, to wówczas można by usunąć wpisy 5* i 4, i zamiast tego rozdzielić wpis 3 na dwa rozłączne.

Przedstawiono to na poniższym obrazku:



Można podzielić 3 na `10.3.0.0/27` (zakres `10.3.0.0` – `10.3.0.31`) i na `10.3.0.128/25` (zakres `10.3.0.128` – `10.3.0.255`).

Teraz możemy usunąć wpisy 5* i 4, gdyż sprawdzany adres może po prostu „wpaść” w powstałą „dziurę” i dopasować się do wpisu 2, który tak jak 5* i 4 kieruje do routera B.

Daje to nam ostateczną tabelę:

	Adresy	Dokąd wysłać	Zakres adresów
1	`0.0.0.0/0`	router A	`0.0.0.0` – `255.255.255.255`
2	`10.0.0.0/8`	router B	`10.0.0.0` – `10.255.255.255`
3*	`10.3.0.0/27`	router C	`10.3.0.0` – `10.3.0.31`
3**	`10.3.0.128/25`	router C	`10.3.0.128` – `10.3.0.255`

Zadanie 5.

Teza. Aby zasada najlepszego dopasowania odpowiadała wyborowi „pierwszy pasujący”, należy uporządkować wpisy malejąco względem długości prefiksów.

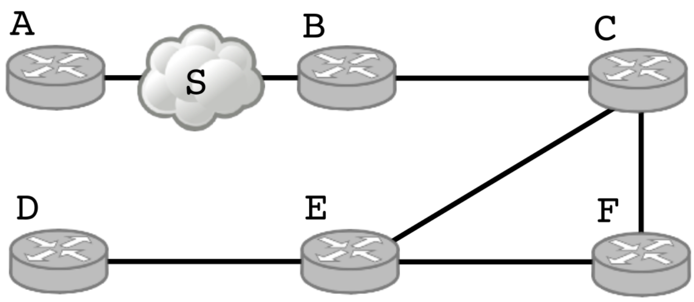
Dowód. Załóżmy, że wpisy są posortowane malejąco względem długości prefiksów. Pokażemy, że wybór pierwszego pasującego wpisu prowadzi do najlepszego dopasowania.

Niech i będzie dowolnym adresem IP i niech k będzie pierwszym w kolejności rozpatrywania wpisem w tablicy routingu, do którego można dopasować i.

Niech n będzie długością prefiksu k. Oznacza to, że pierwsze n bitów i jest identyczne z prefiksem k.

Weźmy dowolny wpis l występujący w tablicy za k. Z założenia, że wpisy posortowane są malejąco względem długości prefiksów, wiemy, że długość m prefiksu l spełnia nierówność $m \leq n$. Oznacza to, że dopasowane zostanie nie więcej bitów adresu i niż n, czyli k jest nie gorszym dopasowaniem niż l, więc wybór „pierwszego pasującego" odpowiada zasadzie najlepszego dopasowania, co należało udowodnić.

Zadanie 6.



Algorytm działa w taki sposób, że sąsiadujące routery co jakiś czas przekazują sobie swoje tablice routingu. Na ich podstawie routery wyznaczają najkrótsze ścieżki metodą relaksacji.

Początkowy stan (każdy zna tylko swoich sąsiadów, puste komórki oznaczają, że droga nie jest znana (długość wynosi nieskończoność)):

	A	**B**	**C**	**D**	**E**	**F**
trasa do A	-	1				
trasa do B	1	-	1			
trasa do C		1	-		1	1
trasa do D				-	1	
trasa do E			1	1	-	1
trasa do F			1		1	-
trasa do S	1	1				

	A	**B**	**C**	**D**	**E**	**F**
trasa do A	-	1	2 (od B)			
trasa do B	1	-	1		2 (od C)	2 (od C)
trasa do C	2 (od B)	1	-	2 (od E)	1	1
trasa do D			2 (od E)	-	1	2 (od E)
trasa do E		2 (od C)	1	1	-	1
trasa do F		2 (od C)	1	2 (od E)	1	-
trasa do S	1	1	2 (od B)			

	A	**B**	**C**	**D**	**E**	**F**
trasa do A	-	1	2 (B)		3 (od C)	3 (od C)
trasa do B	1	-	1	3 (od E)	2 (C)	2 (C)
trasa do C	2 (B)	1	-	2 (E)	1	1
trasa do D		3 (od C)	2 (E)	-	1	2 (E)
trasa do E	3 (od B)	2 (C)	1	1	-	1
trasa do F	3 (od B)	2 (C)	1	2 (E)	1	-
trasa do S	1	1	2 (B)		3 (od C)	3 (od C)

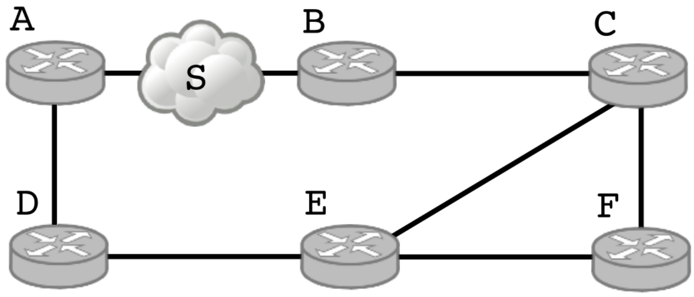
	A	**B**	**C**	**D**	**E**	**F**
trasa do A	-	1	2 (B)	4 (od E)	3 (C)	3 (C)
trasa do B	1	-	1	3 (E)	2 (C)	2 (C)
trasa do C	2 (B)	1	-	2 (E)	1	1
trasa do D	4 (od B)	3 (C)	2 (E)	-	1	2 (E)
trasa do E	3 (B)	2 (C)	1	1	-	1
trasa do F	3 (B)	2 (C)	1	2 (E)	1	-
trasa do S	1	1	2 (B)	4 (od E)	3 (C)	3 (C)

Ostatecznie dostajemy następujące długości dróg:

	A	**B**	**C**	**D**	**E**	**F**
trasa do A	-	1	2 (B)	4 (E)	3 (C)	3 (C)
trasa do B	1	-	1	3 (E)	2 (C)	2 (C)
trasa do C	2 (B)	1	-	2 (E)	1	1
trasa do D	4 (B)	3 (C)	2 (E)	-	1	2 (E)
trasa do E	3 (B)	2 (C)	1	1	-	1
trasa do F	3 (B)	2 (C)	1	2 (E)	1	-
trasa do S	1	1	2 (B)	4 (E)	3 (C)	3 (C)

Jak widać wszystkie routery znają już drogi do wszystkich innych routerów. Można również sprawdzić, że wyznaczone drogi są najkrótszymi w tej sieci, co oznacza, że **stan stabilny został osiągnięty w trzech krokach**.

Zadanie 7. Rozpatrujemy teraz sytuację, w której po utworzeniu tablicy z zadania 6 dodane zostaje nowe połączenie (pomiędzy `A` i `D`). Nasza sieć wygląda zatem tak:



W zerowym kroku routery `A` i `D` muszą się o sobie dowiedzieć. Mamy zatem następującą tablicę:

	A	**B**	**C**	**D**	**E**	**F**
trasa do A	-	1	2 (B)	1 (nowa z A)	3 (C)	3 (C)
trasa do B	1	-	1	3 (E)	2 (C)	2 (C)
trasa do C	2 (B)	1	-	2 (E)	1	1
trasa do D	1 (nowa z D)	3 (C)	2 (E)	-	1	2 (E)
trasa do E	3 (B)	2 (C)	1	1	-	1
trasa do F	3 (B)	2 (C)	1	2 (E)	1	-
trasa do S	1	1	2 (B)	4 (E)	3 (C)	3 (C)

Teraz gdy routery zaczną wymieniać się swoimi tablicami, część z nich będzie mogła skrócić znane sobie ścieżki.

	A	**B**	**C**	**D**	**E**	**F**
trasa do A	-	1	2 (B)	1	2 (od D)	3 (C)
trasa do B	1	-	1	2 (od A)	2 (C)	2 (C)
trasa do C	2 (B)	1	-	2 (E)	1	1
trasa do D	1	2 (od A)	2 (E)	-	1	2 (E)
trasa do E	2 (od D)	2 (C)	1	1	-	1
trasa do F	3 (B)	2 (C)	1	2 (E)	1	-
trasa do S	1	1	2 (B)	2 (od A)	3 (C)	3 (C)

Ostatecznie dostajemy więc następującą tablicę:

	A	**B**	**C**	**D**	**E**	**F**
trasa do A	-	1	2 (B)	1	2 (D)	3 (C)
trasa do B	1	-	1	2 (A)	2 (C)	2 (C)
trasa do C	2 (B)	1	-	2 (E)	1	1
trasa do D	1	2 (A)	2 (E)	-	1	2 (E)
trasa do E	2 (D)	2 (C)	1	1	-	1
trasa do F	3 (B)	2 (C)	1	2 (E)	1	-
trasa do S	1	1	2 (B)	2 (A)	3 (C)	3 (C)

W tym momencie znów osiągnięto stan stabilny, bo wszystkie ścieżki są już najkrótsze i nic nie będzie się zmieniać.