

# Programowanie sieciowe

Tadeusz Trzaskalik

## 8.1. Wprowadzenie

### *Słowa kluczowe*

**Drzewo rozpinające**

**Minimalne drzewo rozpinające**

**Najkrótsza droga w sieci**

**Wierzchołek początkowy**

**Maksymalny przepływ w sieci**

**Źródło**

**Ujście**

## 8.2. Minimalne drzewo rozpinające

### *Sformułowanie zadania*

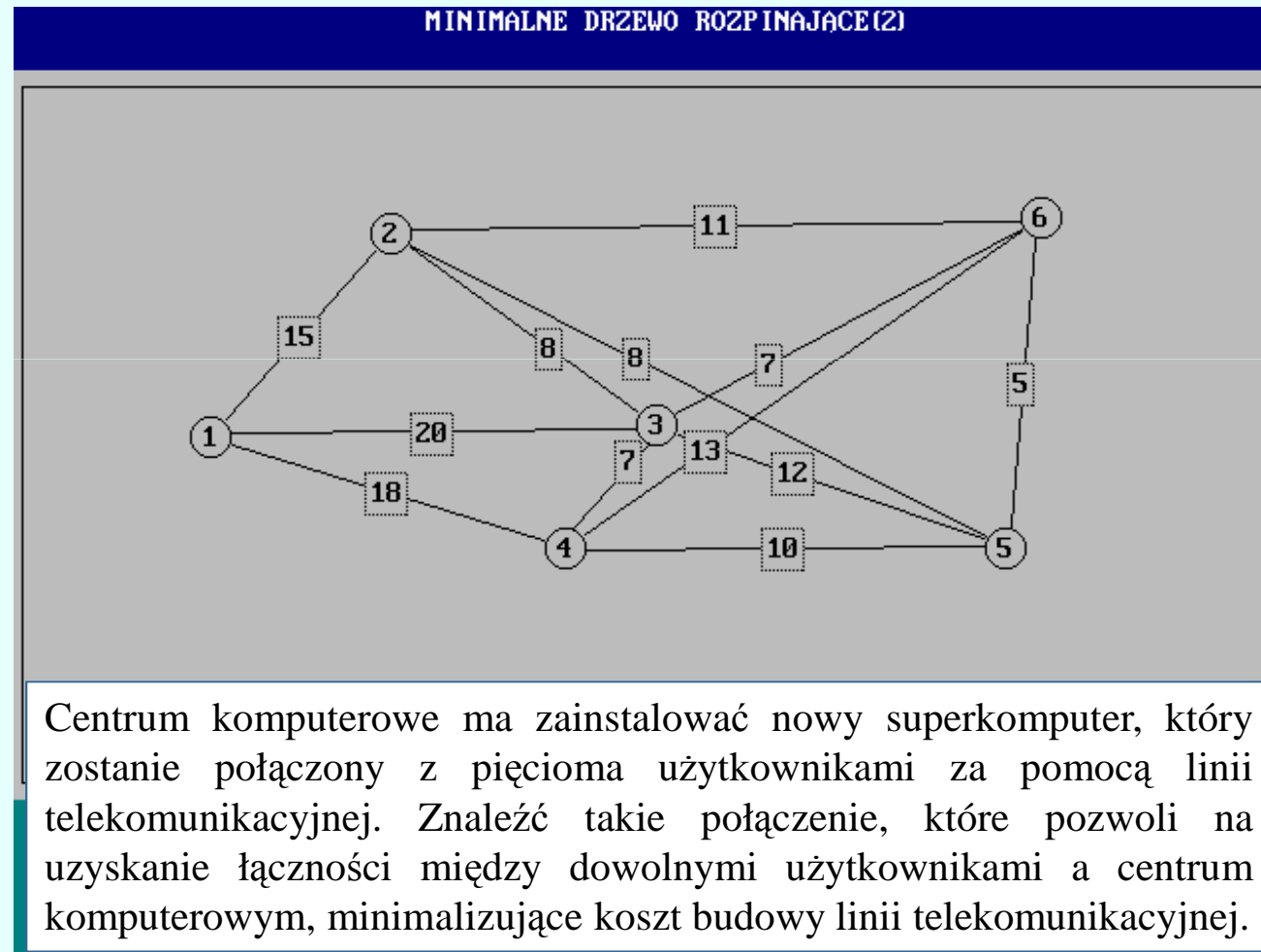
**Drzewo rozpinające** w grafie liczącym  $n$  wierzchołków to zbiór  $n - 1$  jego krawędzi takich, że dowolne dwa wierzchołki grafu można połączyć przy pomocy krawędzi należących do tego zbioru.

**Minimalne drzewo rozpinające** to drzewo wybrane spośród wszystkich istniejących drzew rozpinających, dla którego łączna długość krawędzi jest najmniejsza.

Należy znaleźć minimalne drzewo rozpinające

## 8.2. Minimalne drzewo rozpinające

### Przykład 8.1



## 8.2. Minimalne drzewo rozpinające

### 8.2.1. Reguły postępowania przy poszukiwaniu minimalnego drzewa rozpinającego (1/1)

#### *Algorytm*

#### **Iteracja 1**

Wybieramy w sposób arbitralny dowolny wierzchołek i łączymy go z wierzchołkiem leżącym najbliżej.

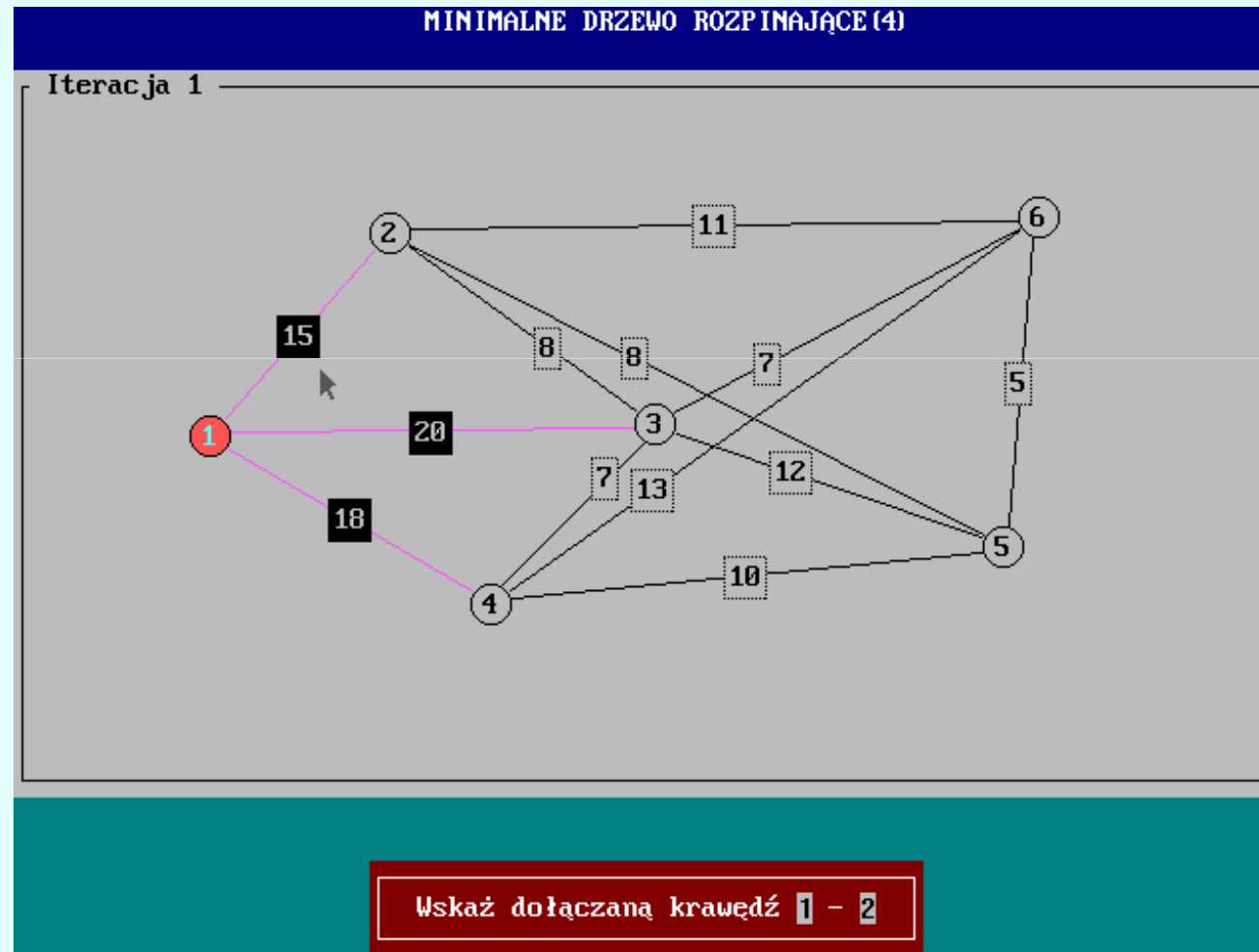
#### **Iteracja $k$ ( $k = 2, \dots, n-1$ )**

W konstruowanym drzewie znajduje się już  $k$  wierzchołków połączonych i  $k-1$  krawędzi. Identyfikujemy najbliższy wierzchołek niepołączony i dołączmy go do zbioru wierzchołków połączonych.

## 8.2. Minimalne drzewo rozpinające

### 8.2.2. Kolejne iteracje (1/6)

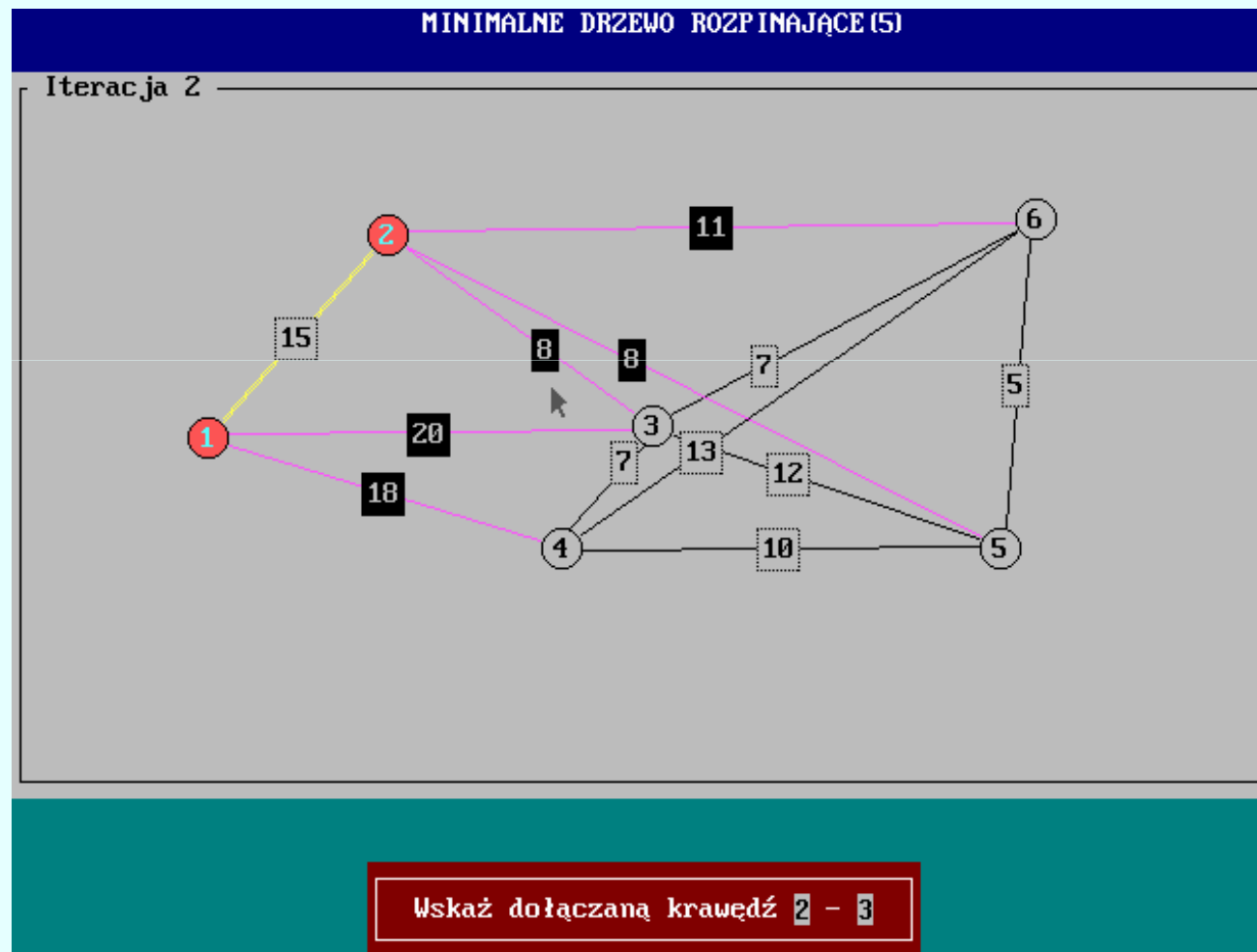
#### Iteracja 1



## 8.2. Minimalne drzewo rozpinające

### 8.2.2. Kolejne iteracje (2/6)

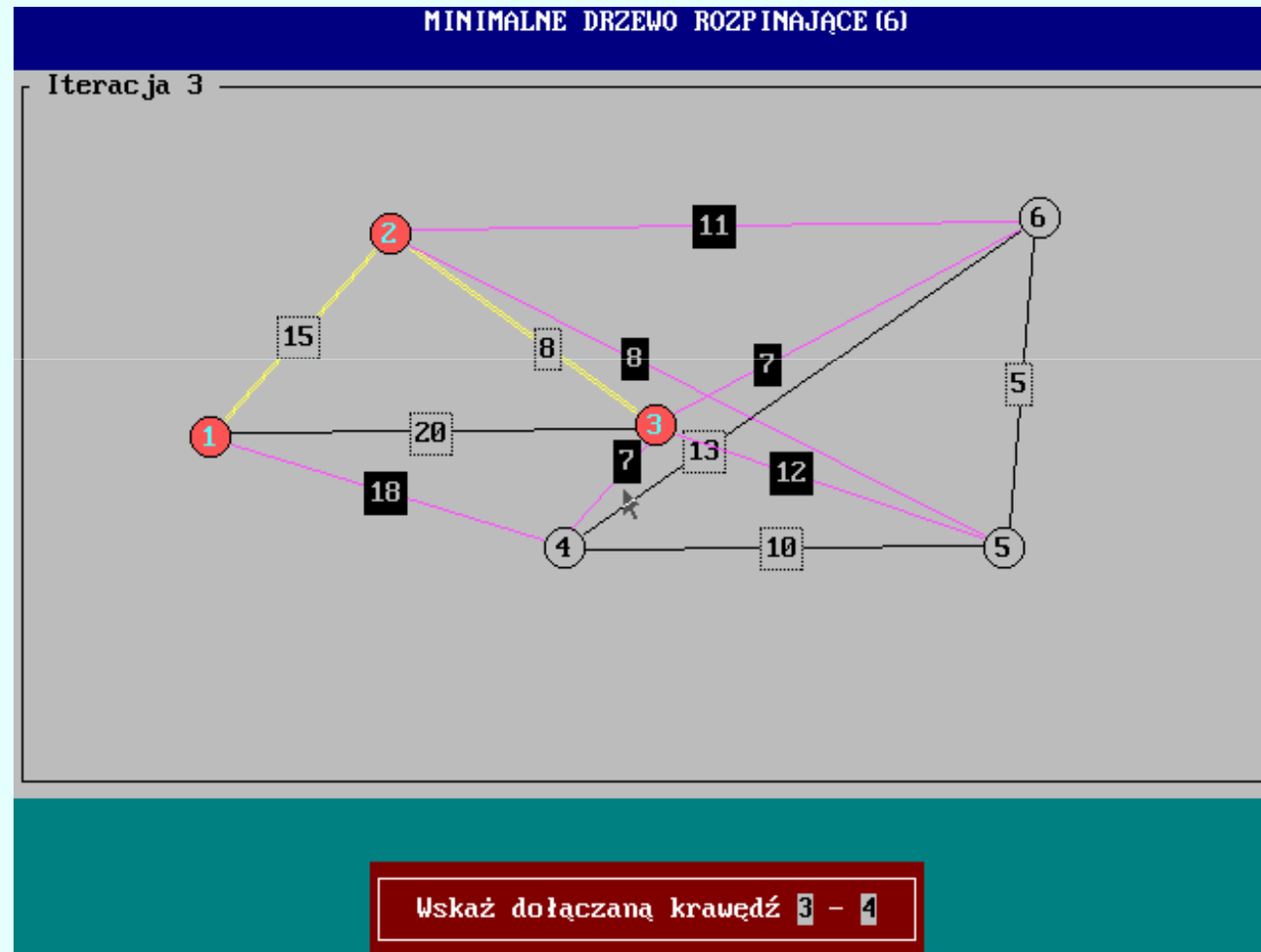
#### Iteracja 2



## 8.2. Minimalne drzewo rozpinające

### 8.2.2. Kolejne iteracje (3/6)

#### Iteracja 3

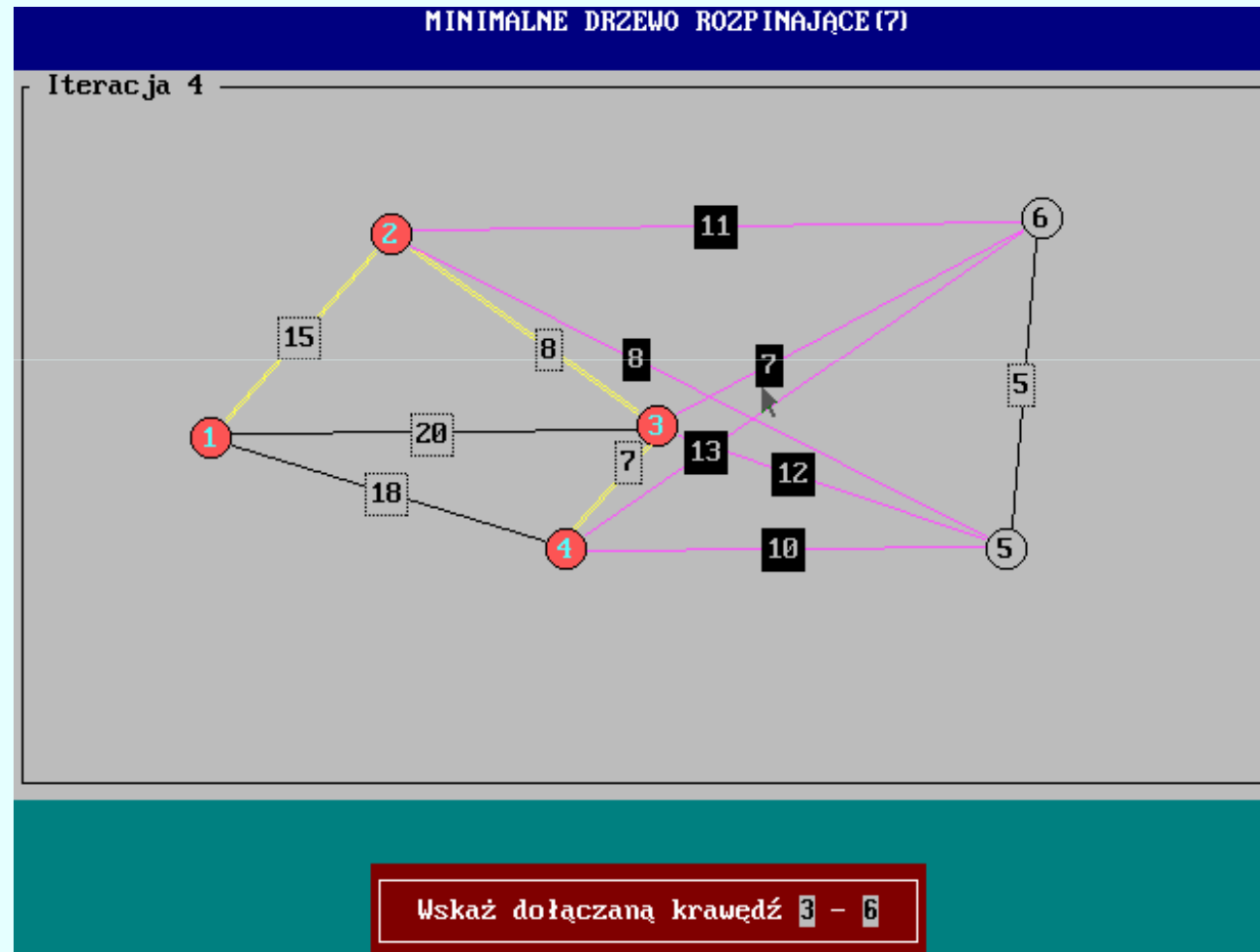




## 8.2. Minimalne drzewo rozpinające

### 8.2.2. Kolejne iteracje (4/6)

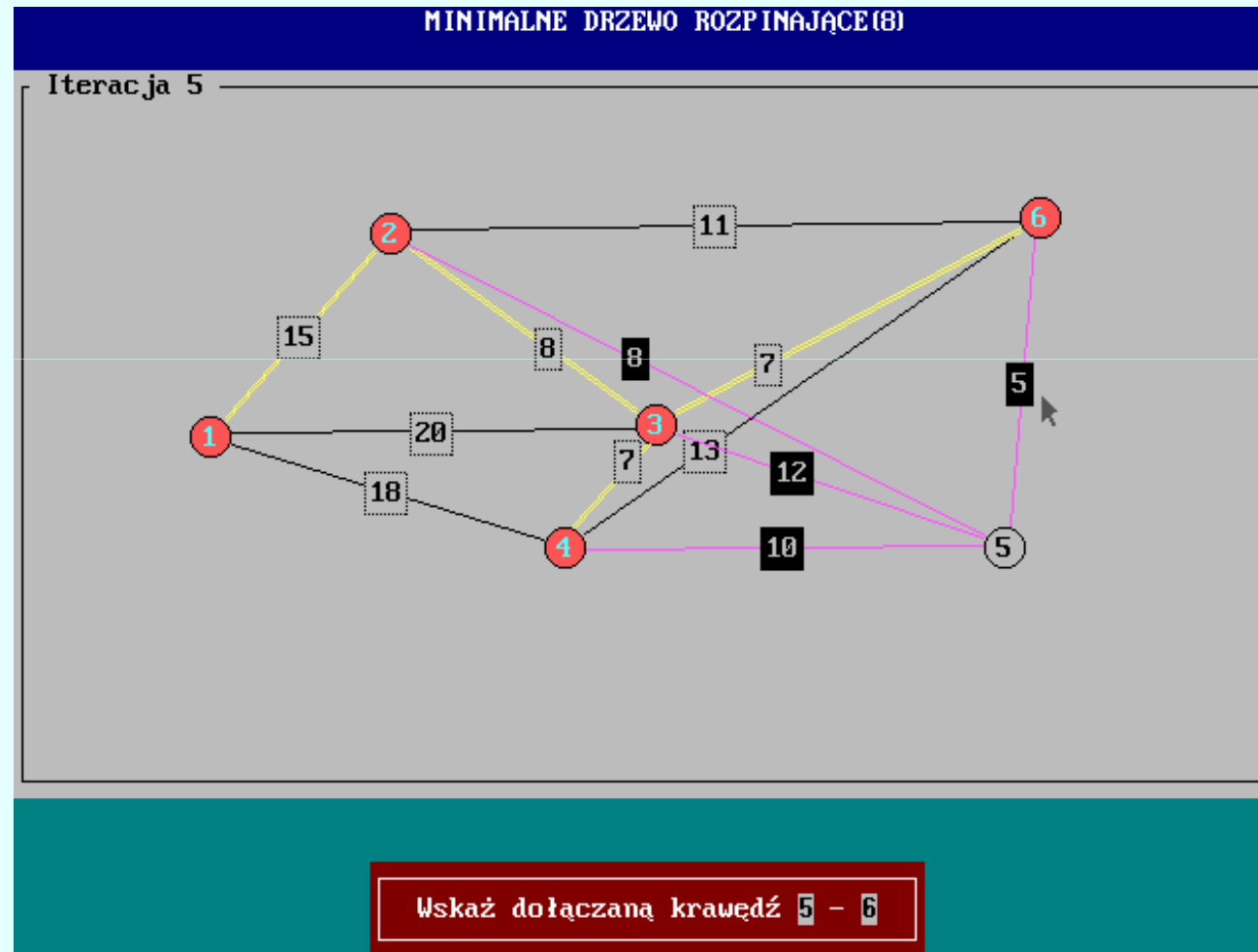
#### Iteracja 4



## 8.2. Minimalne drzewo rozpinające

### 8.2.2. Kolejne iteracje (5/6)

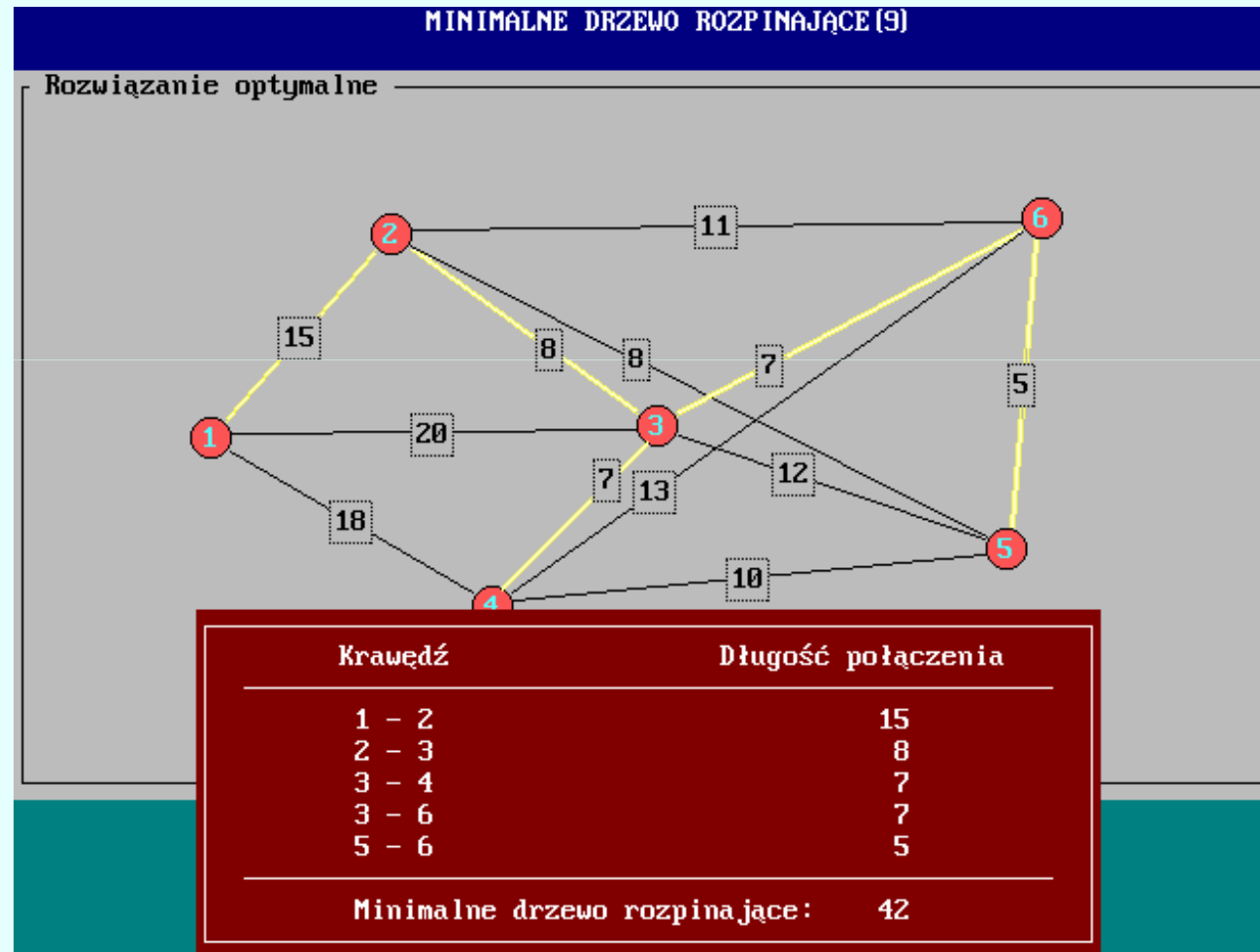
#### Iteracja 5



## 8.2. Minimalne drzewo rozpinające

### 8.2.2. Kolejne iteracje (6/6)

#### Rozwiązanie optymalne



## 8.3. Najkrótsze drogi w sieci

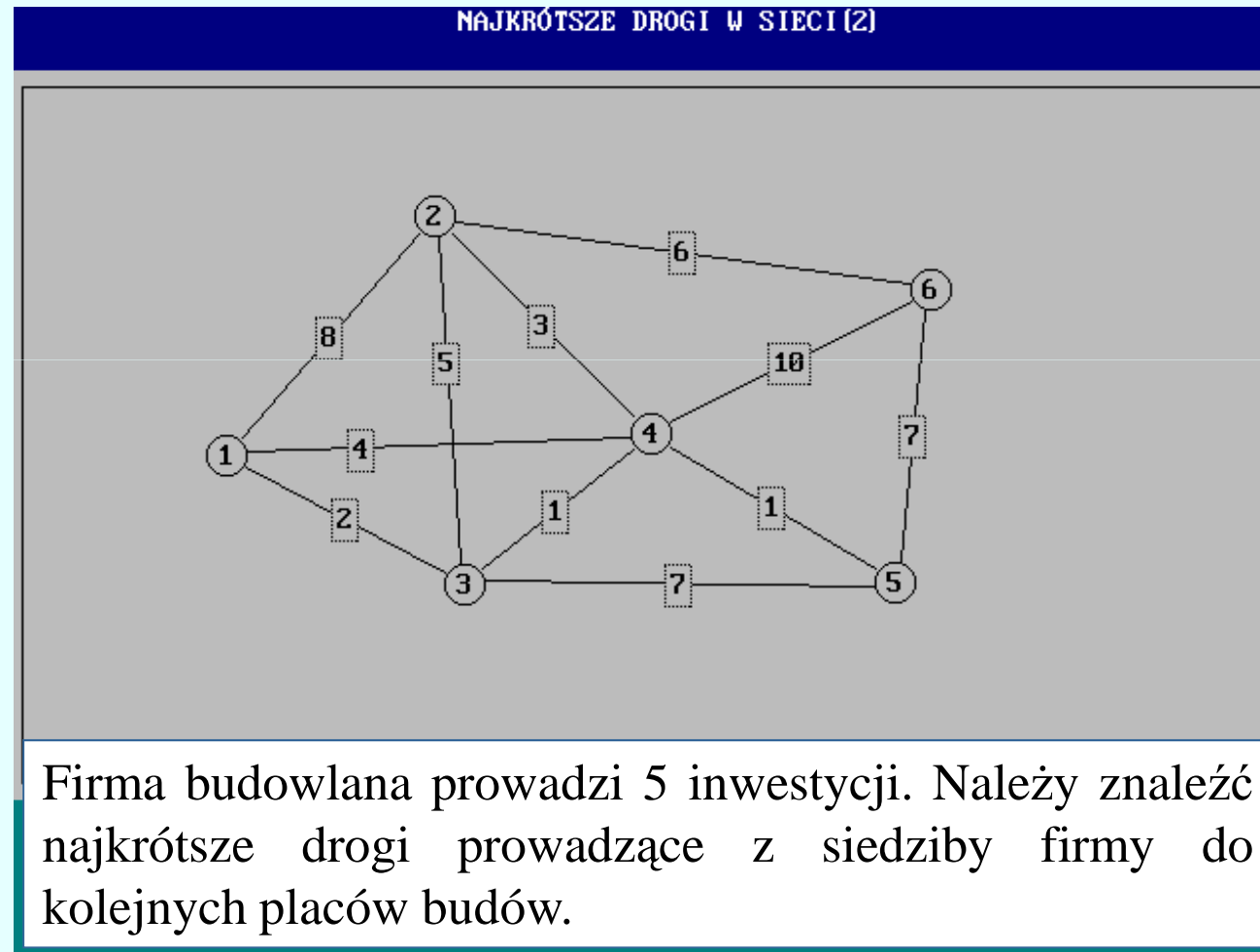
### *Sformułowanie zadania*

**Najkrótsza droga** pomiędzy dwoma dowolnie wybranymi wierzchołkami w sieci to taki zbiór krawędzi łączących te wierzchołki, dla których suma długości jest najmniejsza.

Należy znaleźć najkrótsze drogi łączące wierzchołek początkowy (numer 1) ze wszystkimi pozostałymi wierzchołkami grafu.

## 8.3. Najkrótsze drogi w sieci

### Przykład 8.2



## 8.3. Najkrótsze drogi w sieci

### 8.3.1. Reguły postępowania przy poszukiwaniu najkrótszych dróg w sieci (1/3)

#### *Algorytm*

#### Faza 1. Cechowanie wierzchołków.

#### **Iteracja 1.**

Wierzchołkiem cechowanym na stałe jest wierzchołek (1). Przyporządkowujemy mu etykietę stałą  $[0, S]$  Znajdujemy krawędzie rozpoczynające się w wierzchołku początkowym. Wierzchołkom kończącym te krawędzie przyporządkowujemy etykiety tymczasowe

## 8.3. Najkrótsze drogi w sieci

### 8.3.1. Reguły postępowania przy poszukiwaniu najkrótszych dróg w sieci (2/3)

#### *Algorytm (c.d.)*

#### **Iteracja $k$ ( $k \leq n$ ).**

Z wierzchołków odcachowanych tymczasowo wybieramy wierzchołek, który cechujemy na stałe. Jest to ten wierzchołek, który ma pierwszą składową najmniejszą. Znajdujemy krawędzie prowadzące od wybranego wierzchołka do wierzchołków, które nie zostały dotychczas odcachowane na stałe i nadajemy im etykiety tymczasowe lub modyfikujemy istniejące

## 8.3. Najkrótsze drogi w sieci

### 8.3.1. Reguły postępowania przy poszukiwaniu najkrótszych dróg w sieci (3/3)

*Algorytm (c.d.)*

#### **Faza 2. Identyfikacja najkrótszych dróg.**

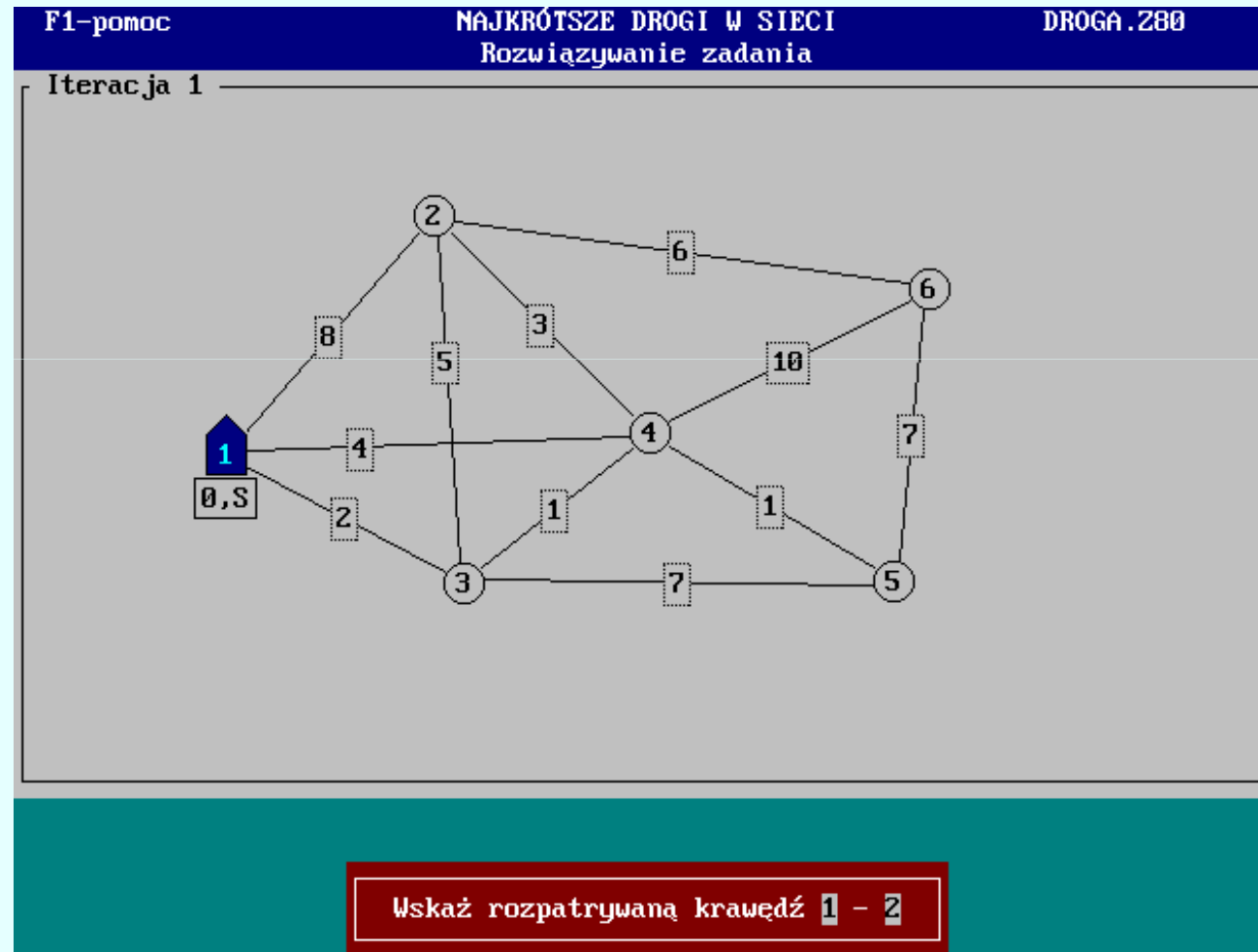
Dla kolejnych wierzchołków, rozpoczynając od wierzchołka 2 identyfikujemy krawędzie, wchodzące w skład najkrótszej drogi. Do tego celu służy druga składowa etykiety stałej.



## 8.3. Najkrótsze drogi w sieci

### 8.3.2. Kolejne iteracje (1/7)

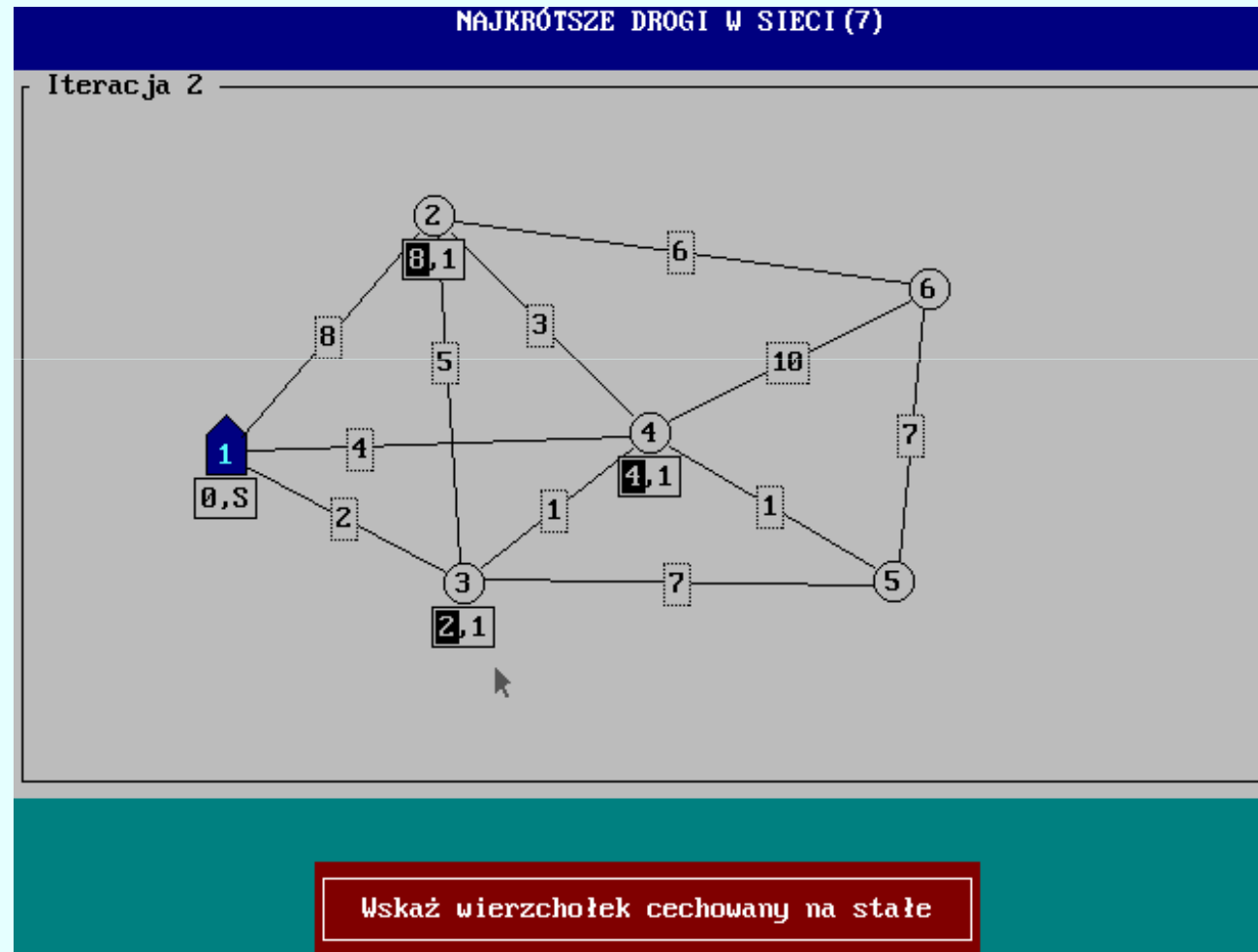
#### Iteracja 1



## 8.3. Najkrótsze drogi w sieci

### 8.3.2. Kolejne iteracje (2/7)

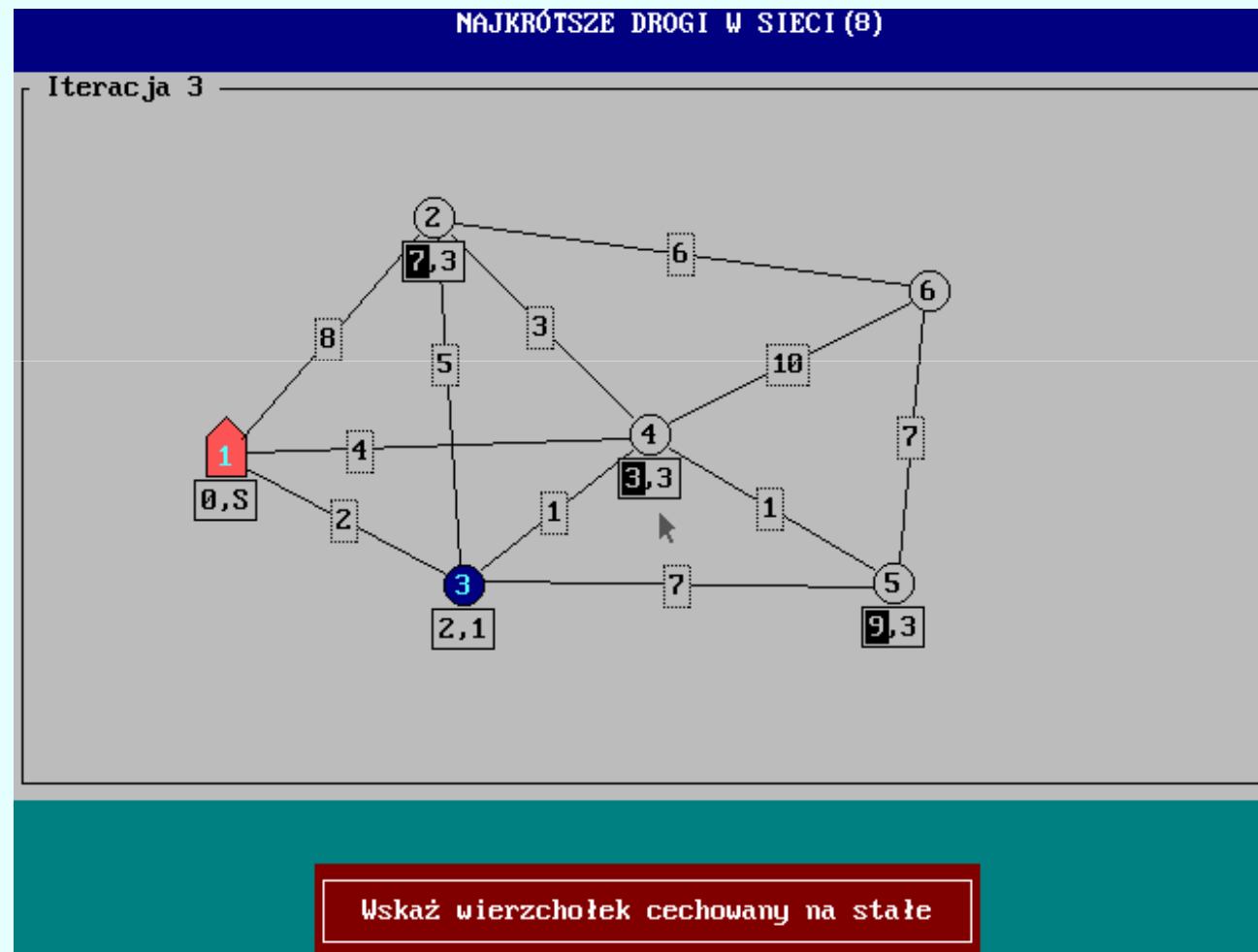
#### Iteracja 2



## 8.3. Najkrótsze drogi w sieci

### 8.3.2. Kolejne iteracje (3/7)

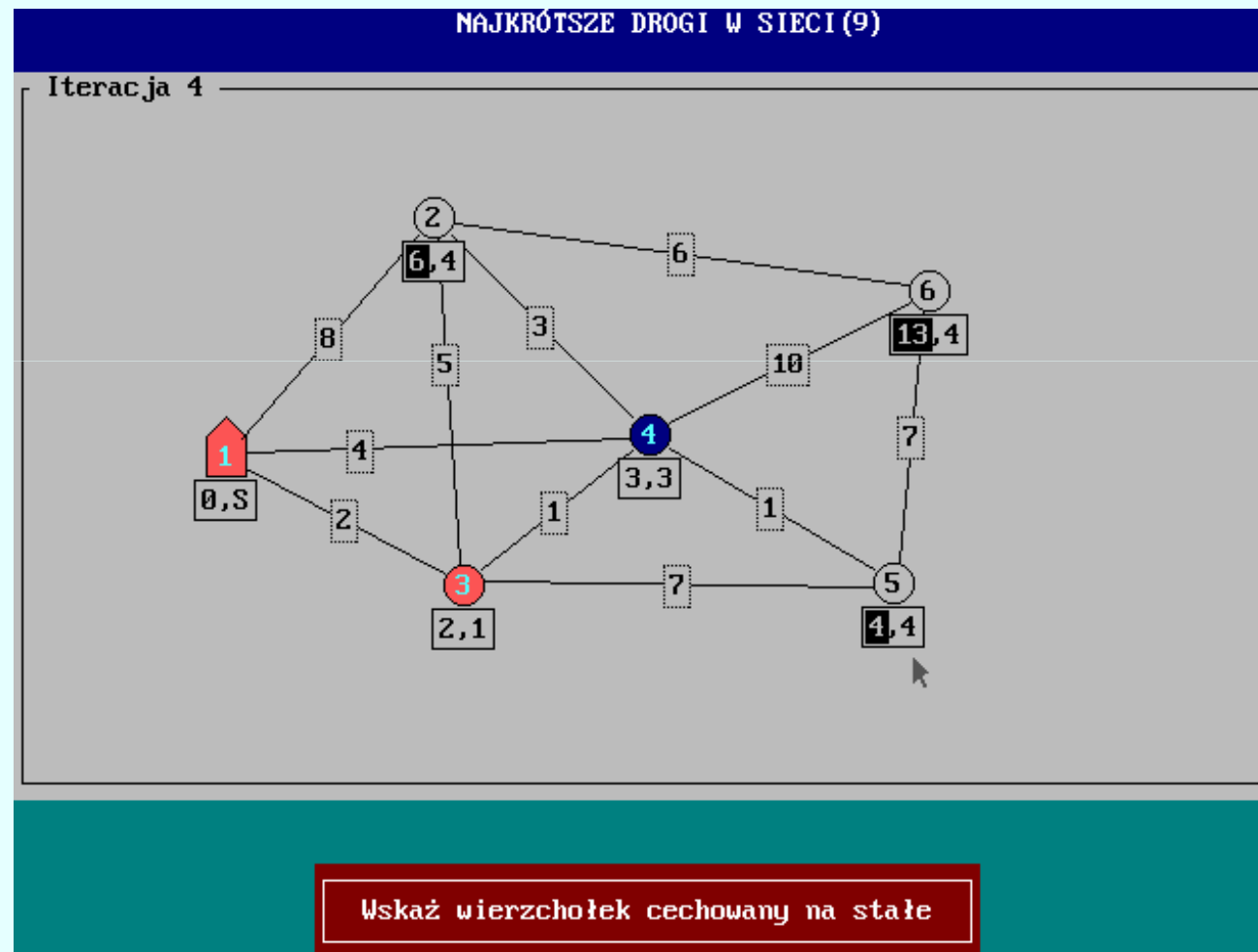
#### Iteracja 3



## 8.3. Najkrótsze drogi w sieci

### 8.3.2. Kolejne iteracje (4/7)

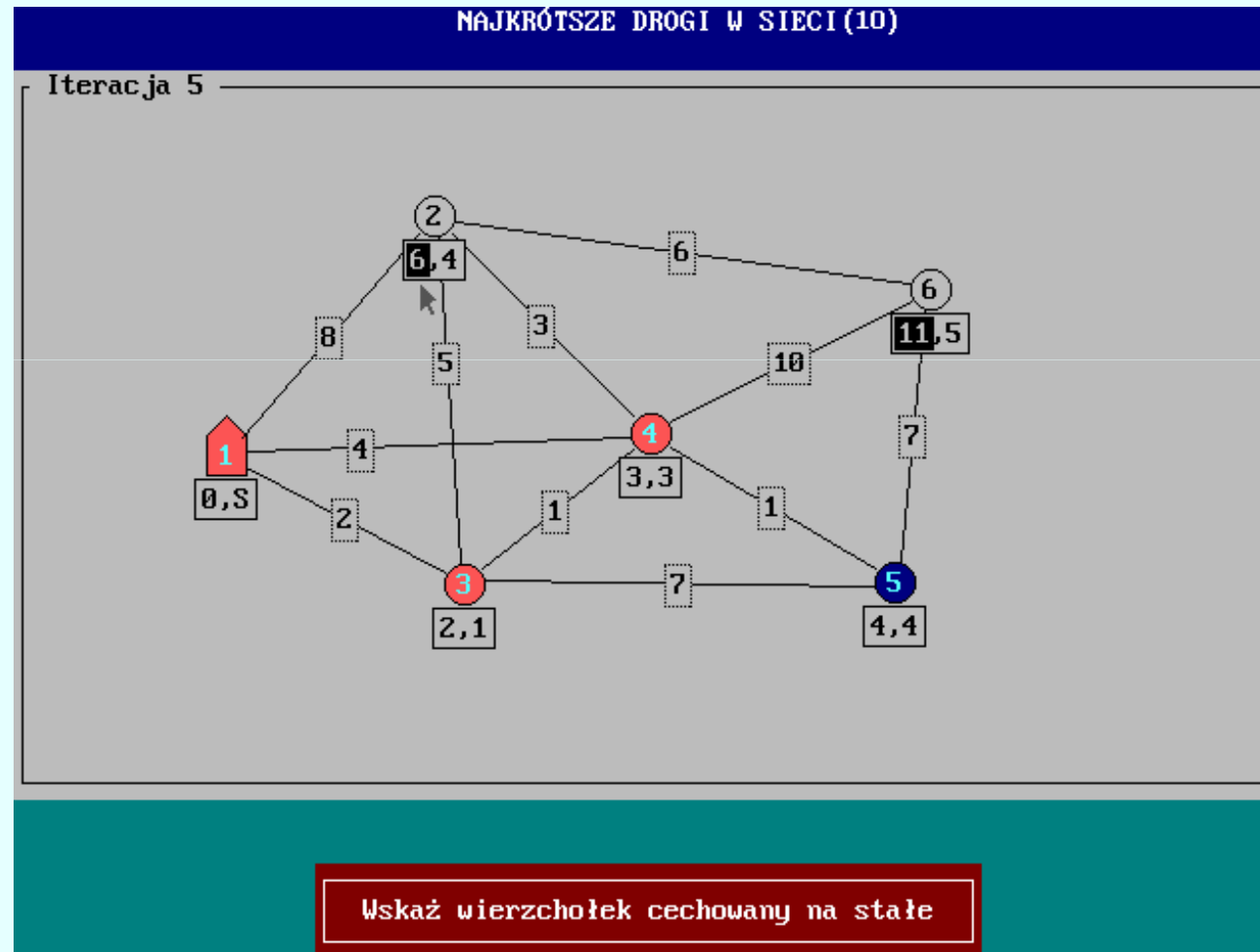
#### Iteracja 4



## 8.3. Najkrótsze drogi w sieci

### 8.3.2. Kolejne iteracje (5/7)

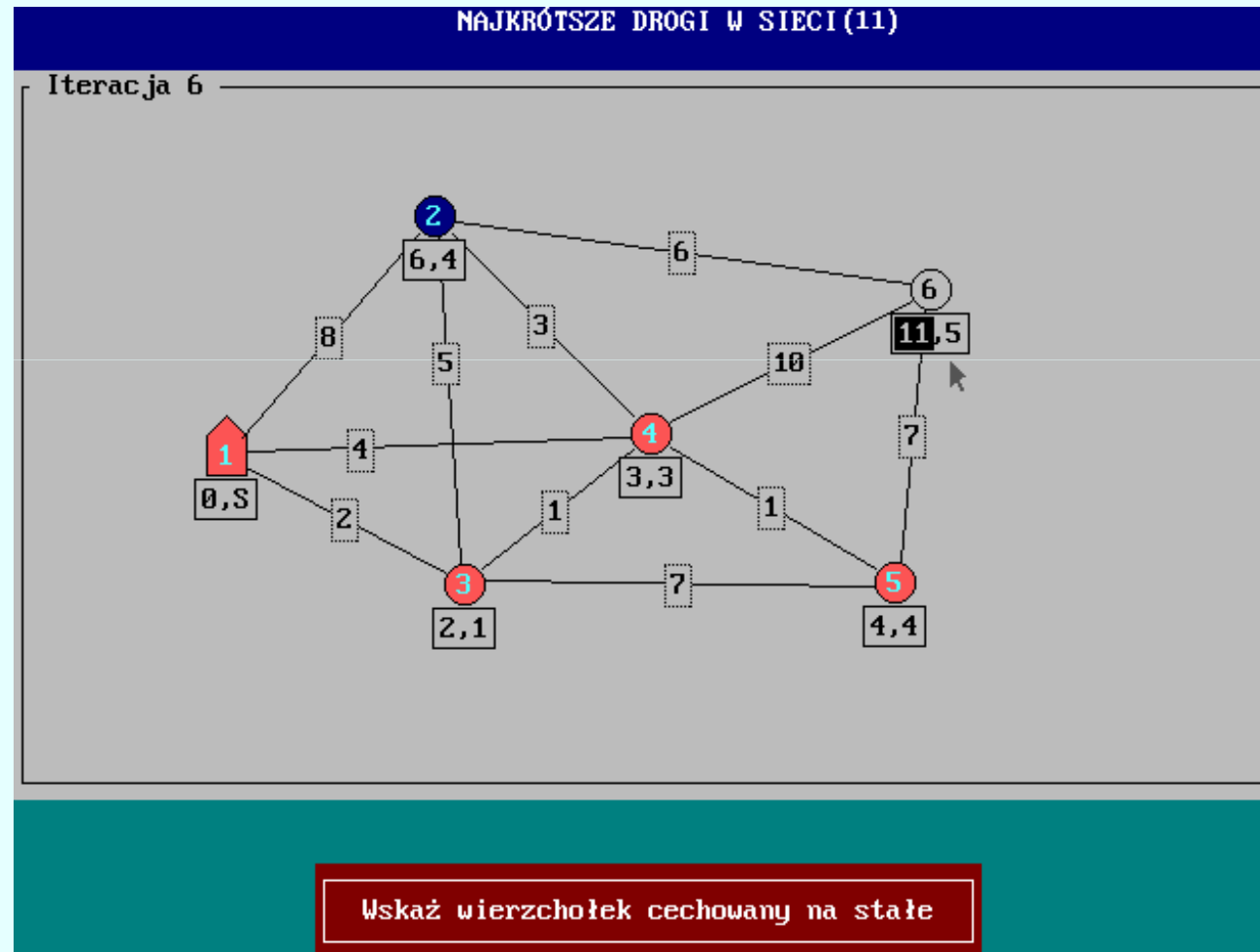
#### Iteracja 5



## 8.3. Najkrótsze drogi w sieci

### 8.3.2. Kolejne iteracje (6/7)

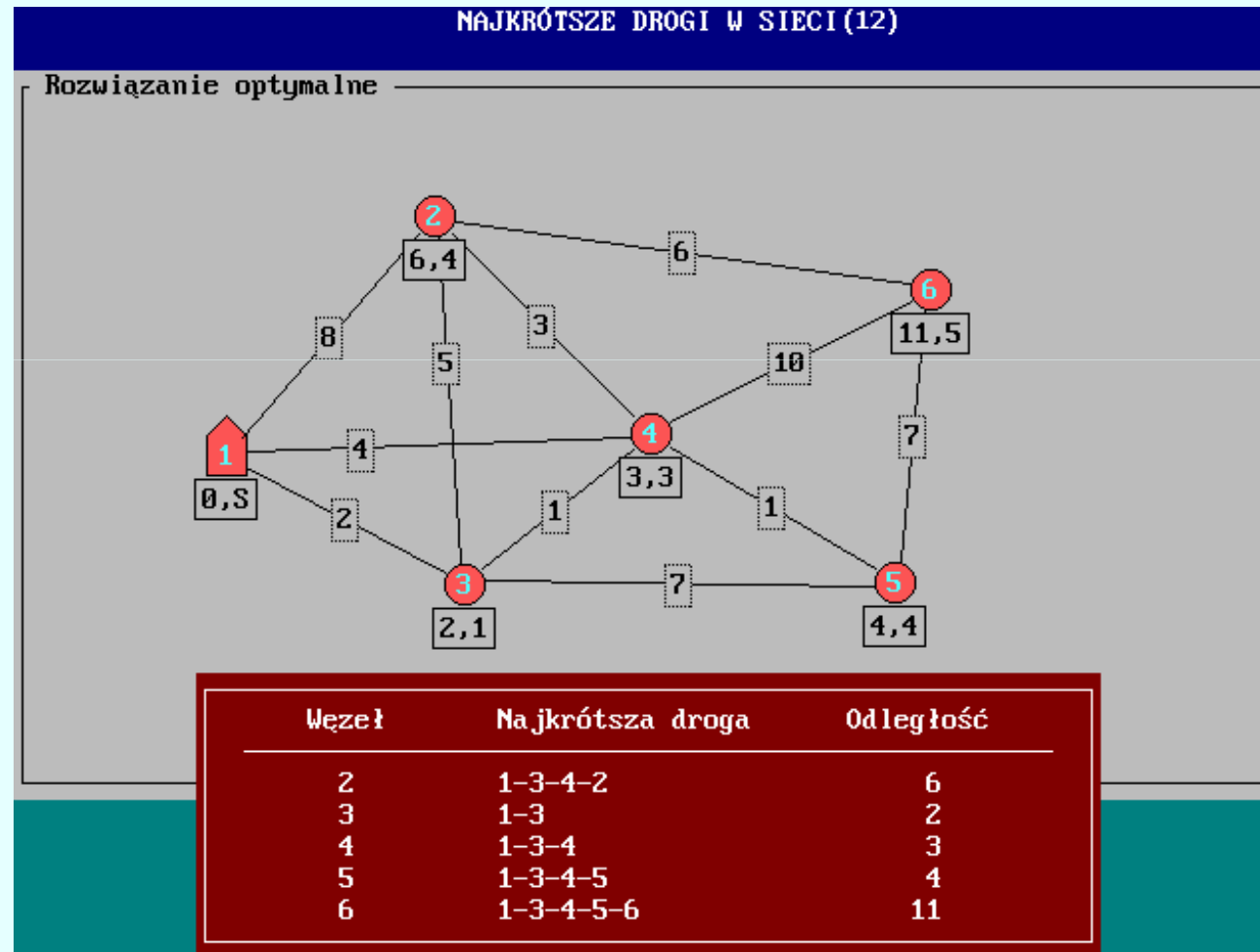
#### Iteracja 6



## 8.3. Najkrótsze drogi w sieci

### 8.3.2. Kolejne iteracje (7/7)

#### Rozwiązanie optymalne



## 8.4. Maksymalny przepływ w sieci

### *Sformułowanie zadania*

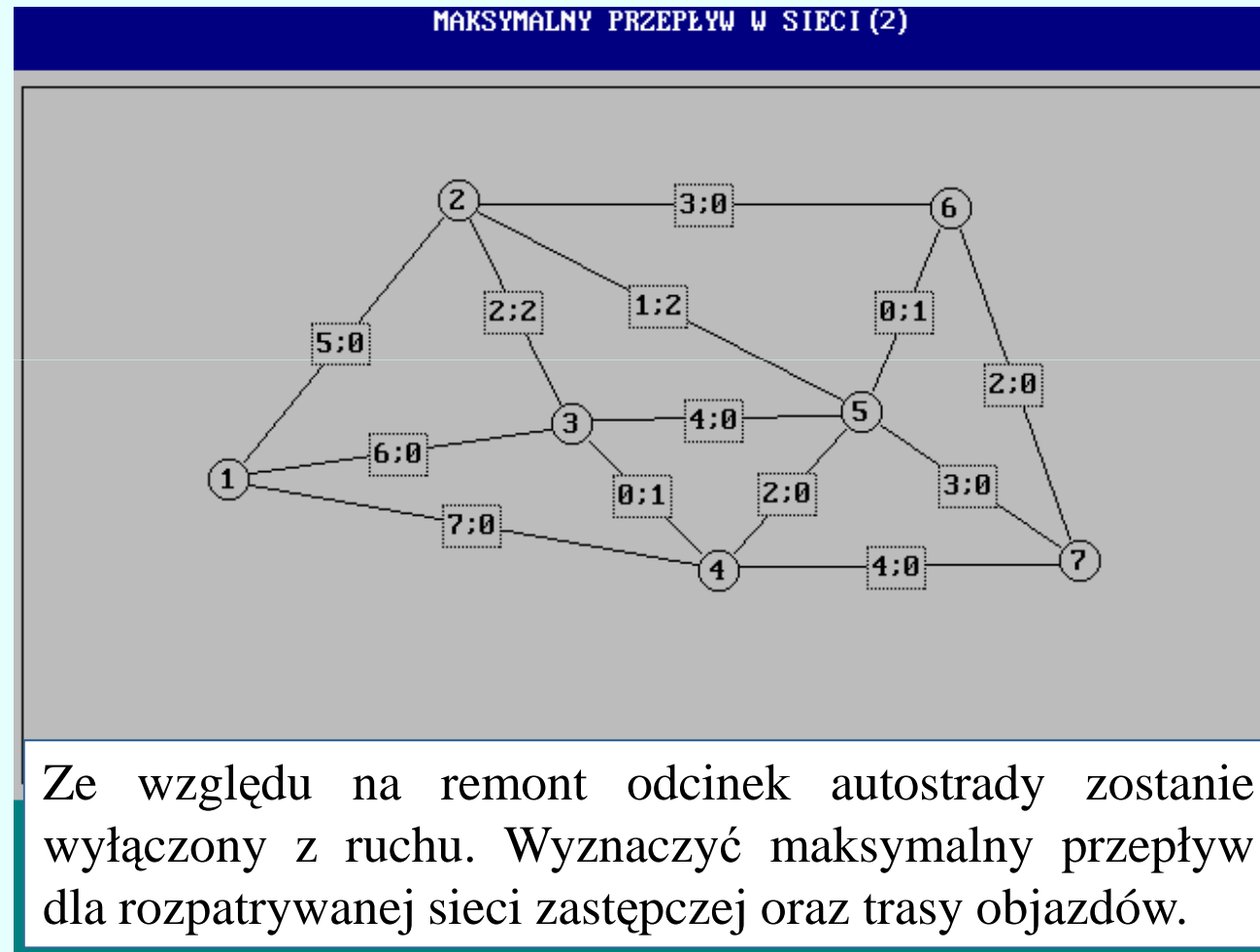
Wyróżniamy wierzchołek początkowy, zwany **źródłem** oraz wierzchołek końcowy, zwany **ujściem**. Każda krawędź grafu opisana jest dwoma liczbami, charakteryzującymi przepustowość krawędzi

Zaplanować przepływ pomiędzy źródłem i ujściem, uwzględniając przepustowość wszystkich krawędzi, by łączna wielkość tego przepływu była największa



## 8.4. Maksymalny przepływ w sieci

### Przykład 8.3



## 8.4. Maksymalny przepływ w sieci

### 8.4.1. Reguły postępowania przy poszukiwaniu maksymalnego przepływu w sieci (1/1)

#### *Algorytm*

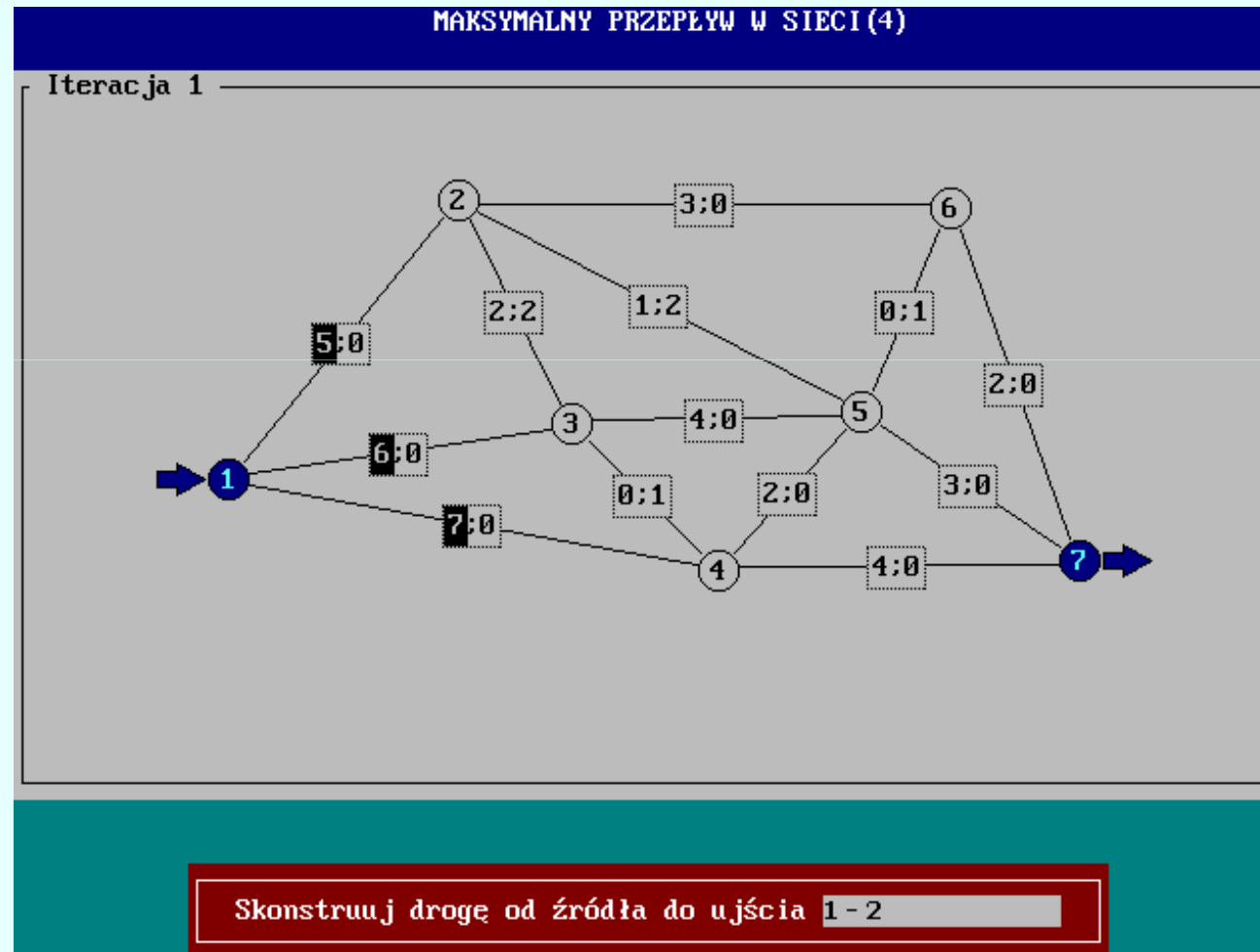
W każdej iteracji wykonujemy następujące kroki:

1. Znajdujemy dowolną ścieżkę prowadzącą od źródła do ujścia, dla której przepustowości wszystkich łuków są większe od zera. O ile takiej ścieżki nie można wyznaczyć, to przepływ nie istnieje (lub maksymalny przepływ wyznaczono już w poprzednich krokach).
2. Na wyznaczonej drodze znajdujemy krawędź o najmniejszej przepustowości  $p$  w rozpatrywanym kierunku.
3. Zmniejszamy przepustowość łuków leżących na znalezionej ścieżce o  $p$  i zwiększamy o tę samą wartość odpowiednie przepustowości residualne. Wracamy do kroku 1.

## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (1/9)

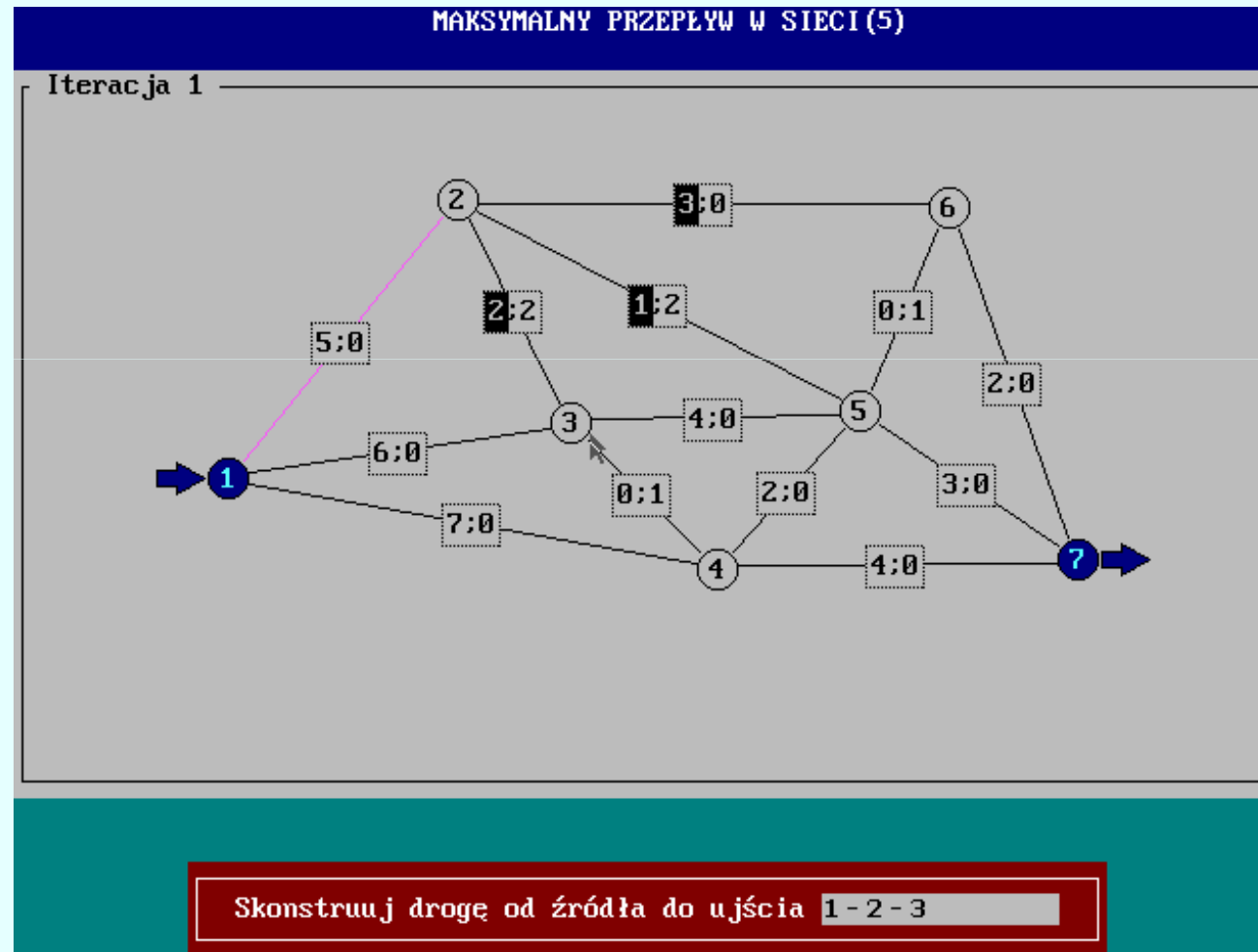
#### Iteracja 1



## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (2/9)

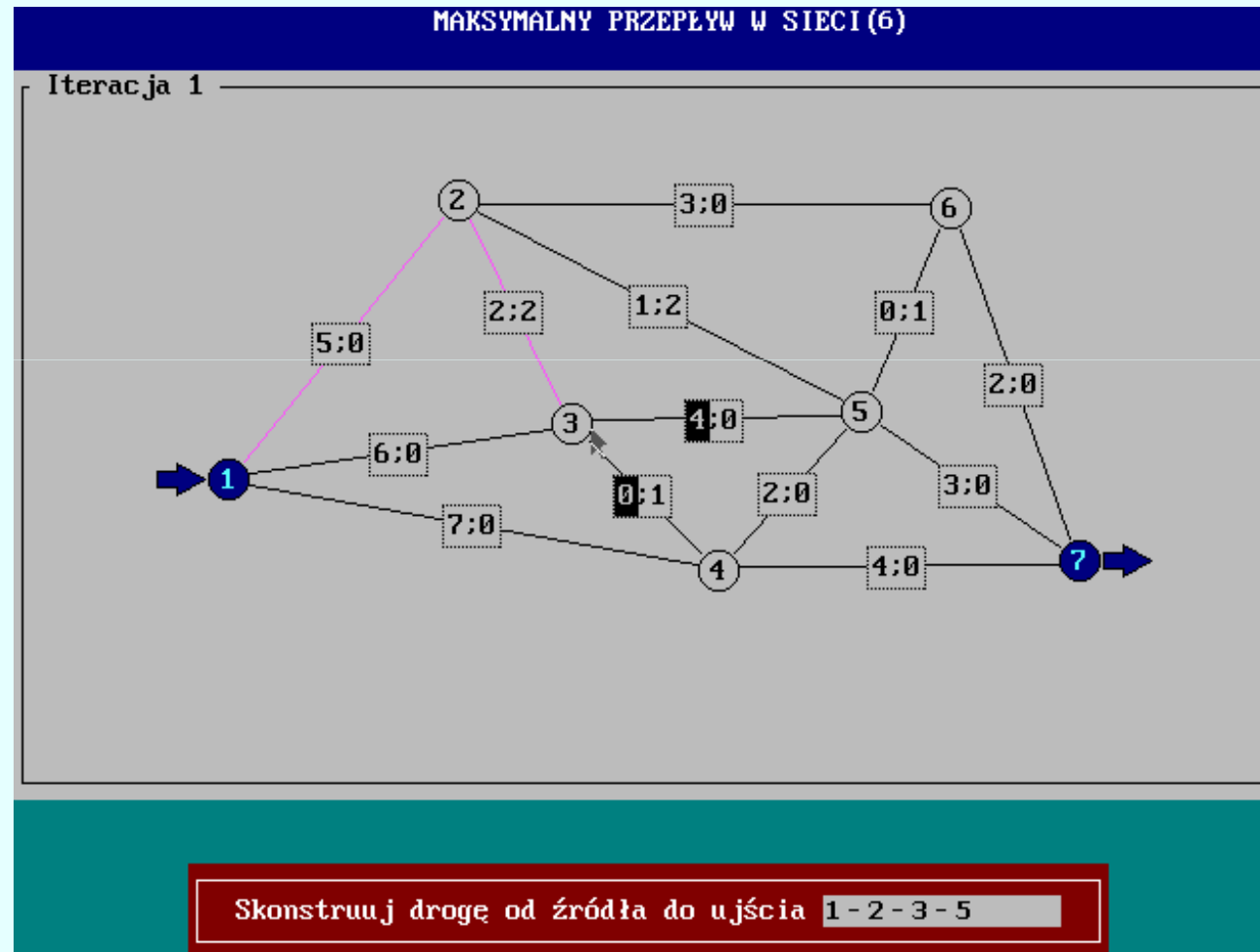
#### Iteracja 1 (c.d.)



## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (3/9)

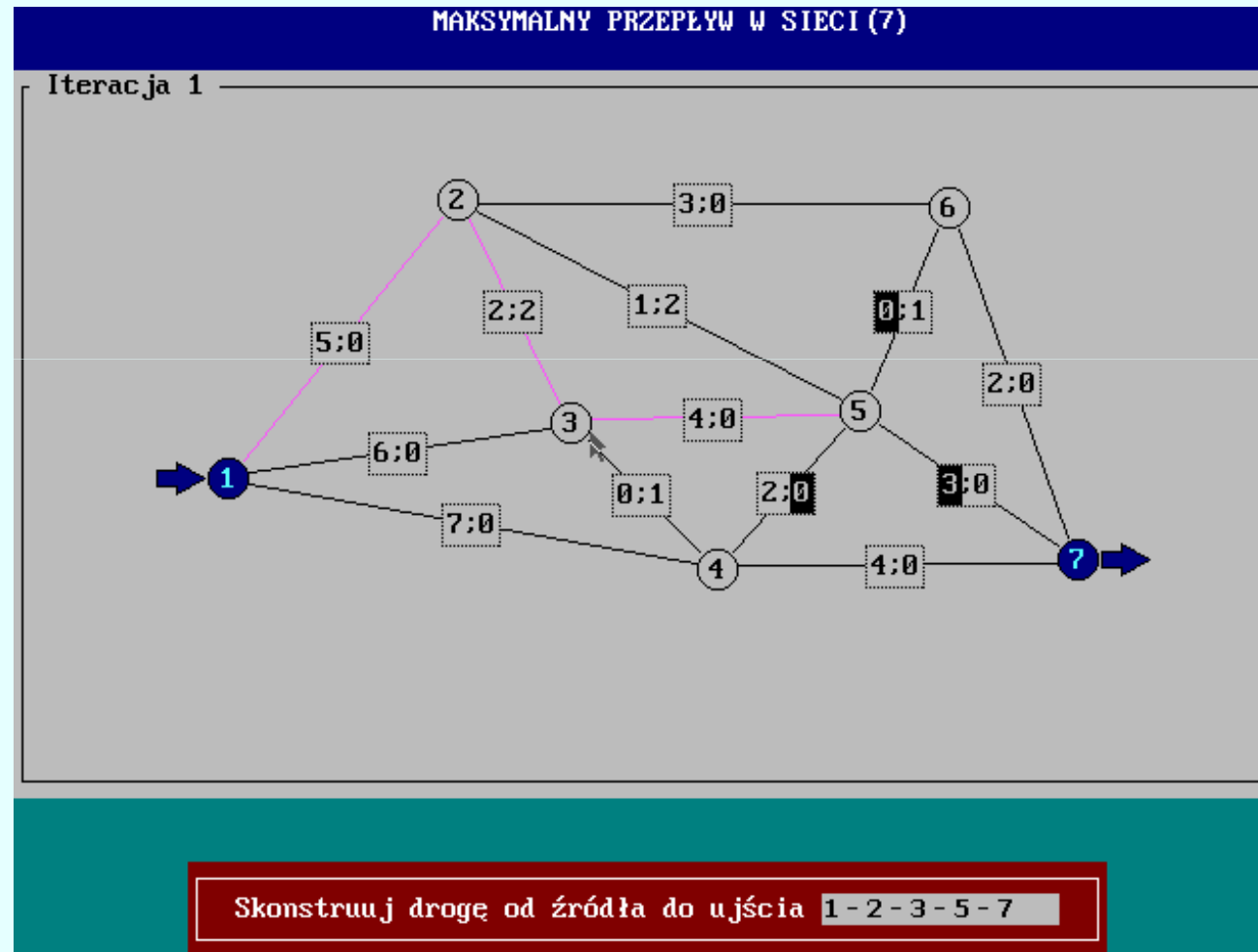
#### Iteracja 1 (c.d.)



## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (4/9)

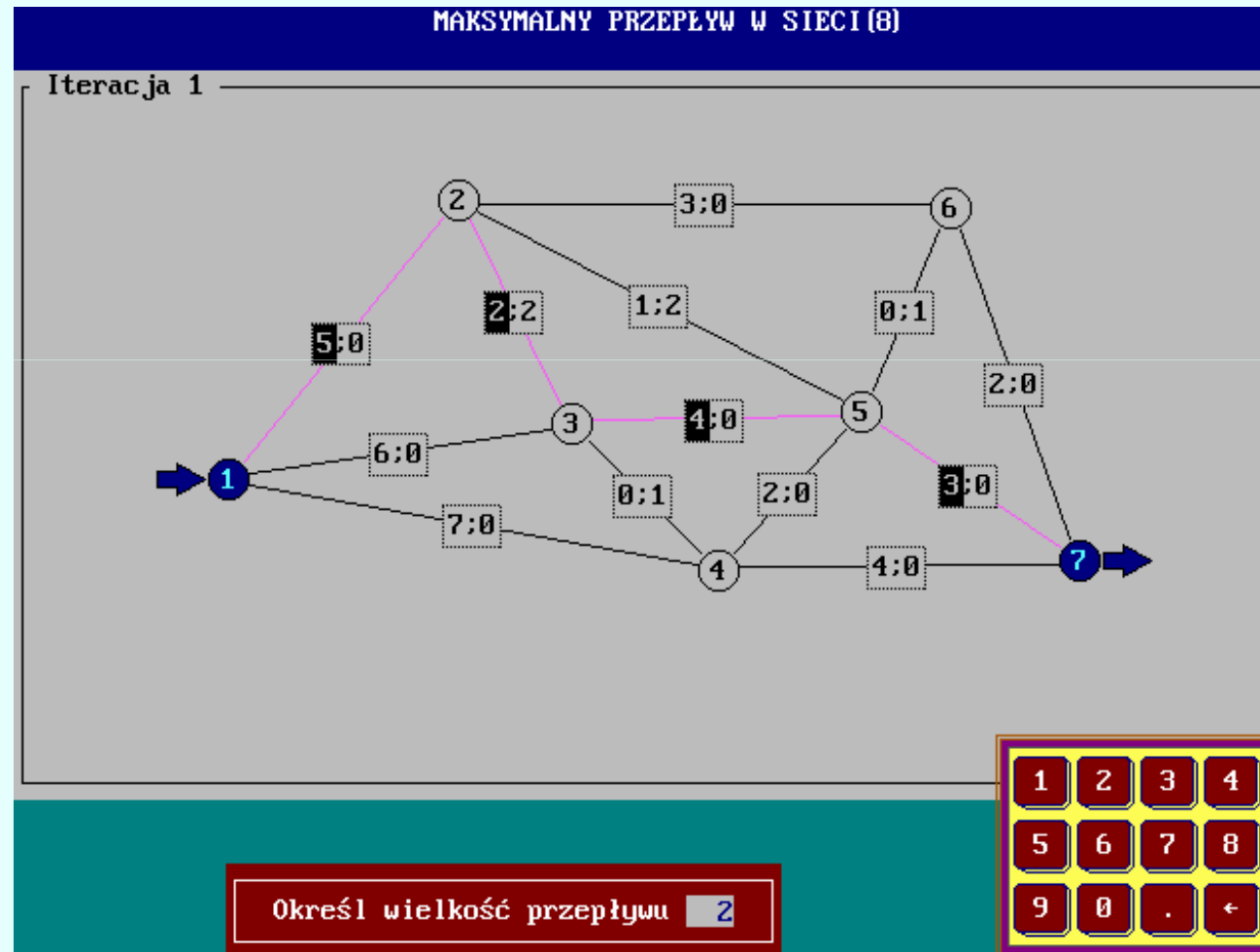
#### Iteracja 1 (c.d.)



## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (5/9)

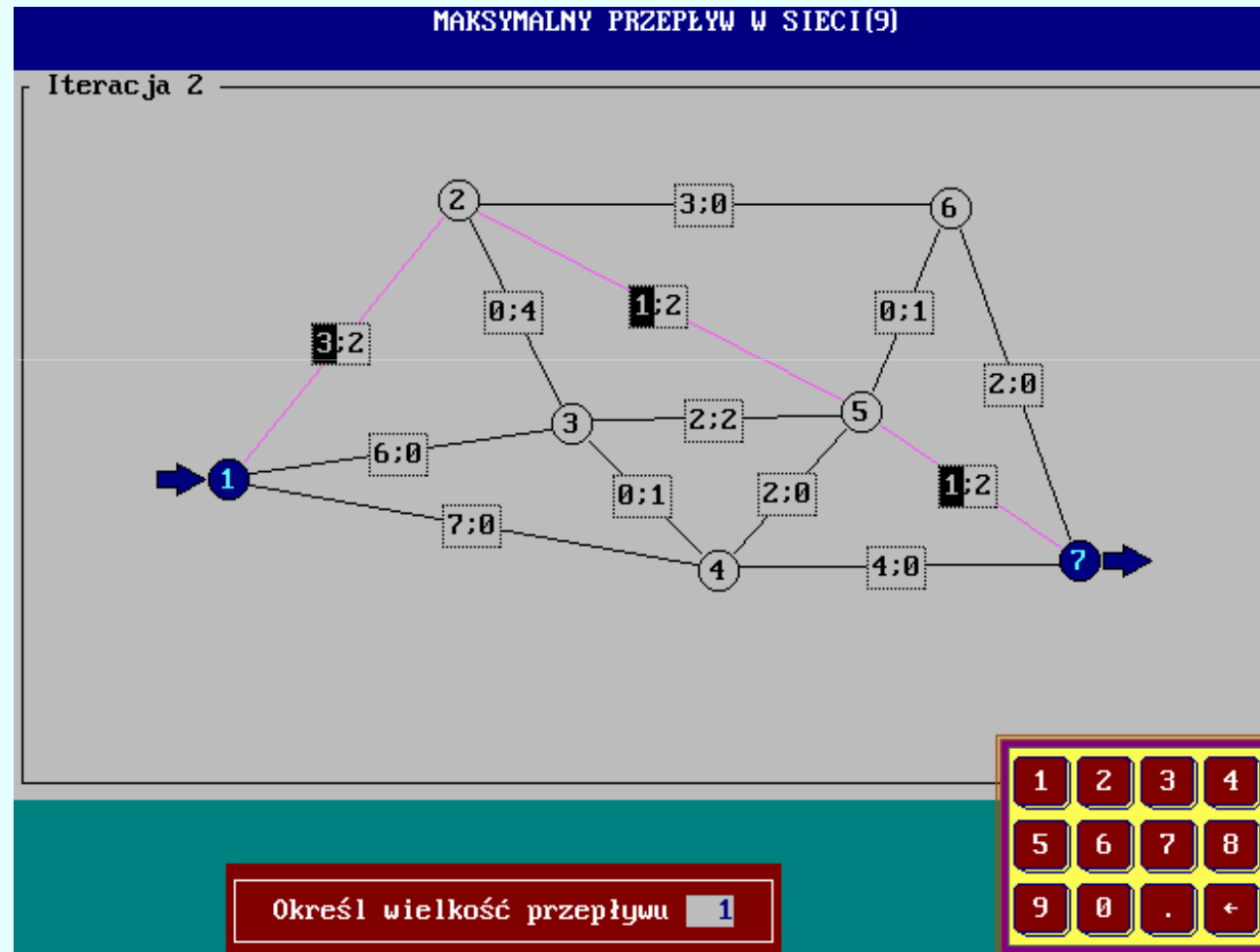
#### Iteracja 1 (c.d.)



## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (6/9)

#### Iteracja 2

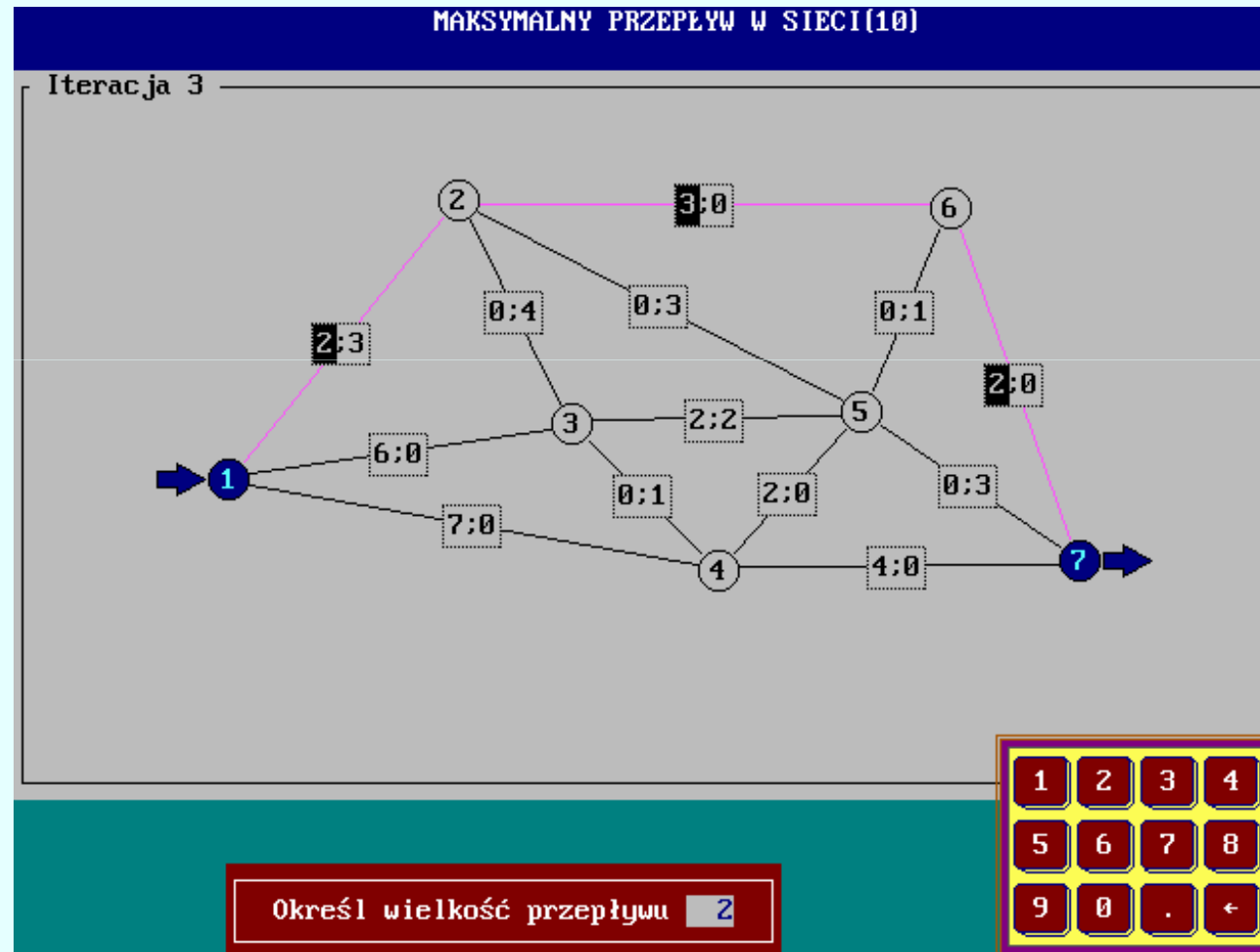




## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (7/9)

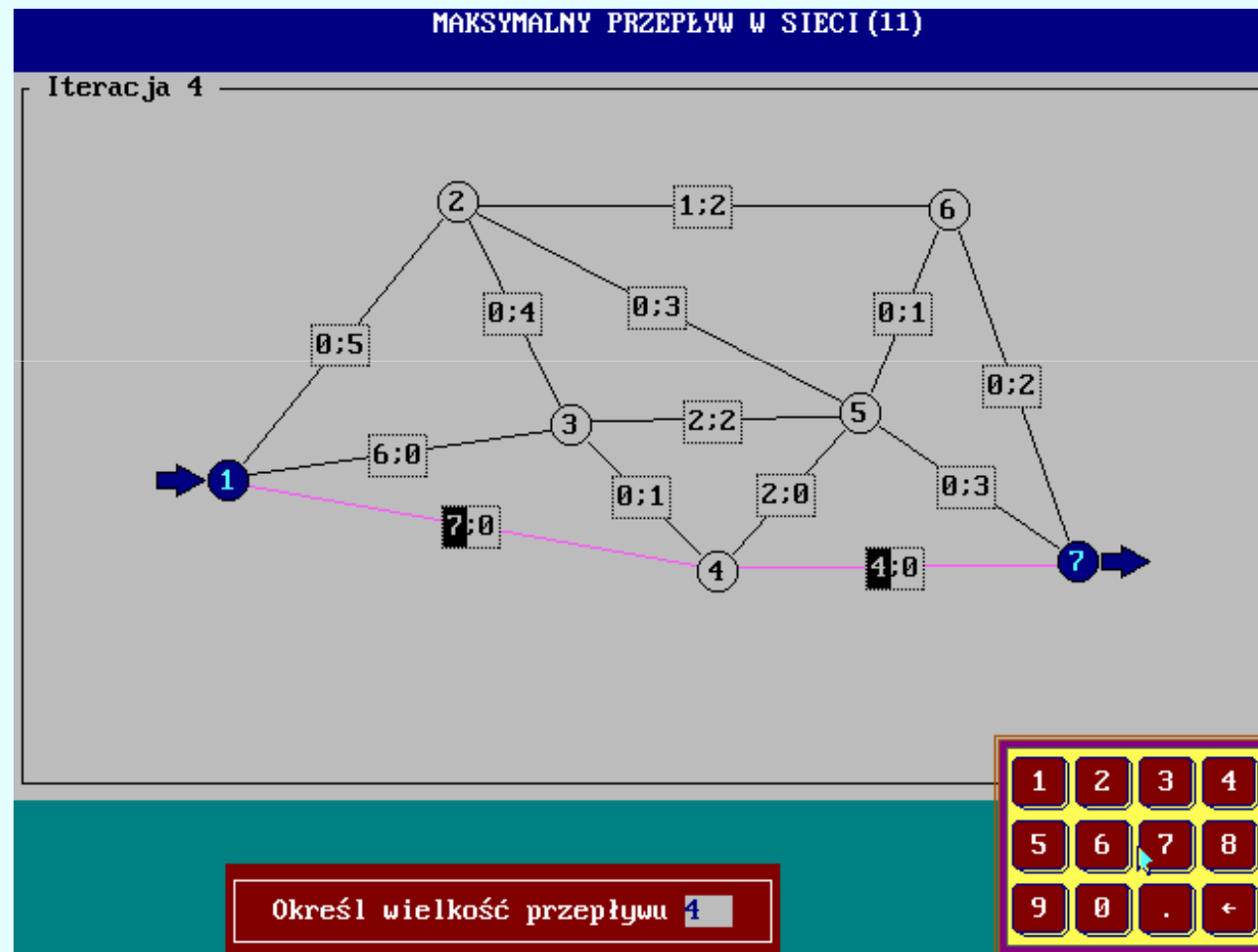
#### Iteracja 3



## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (8/9)

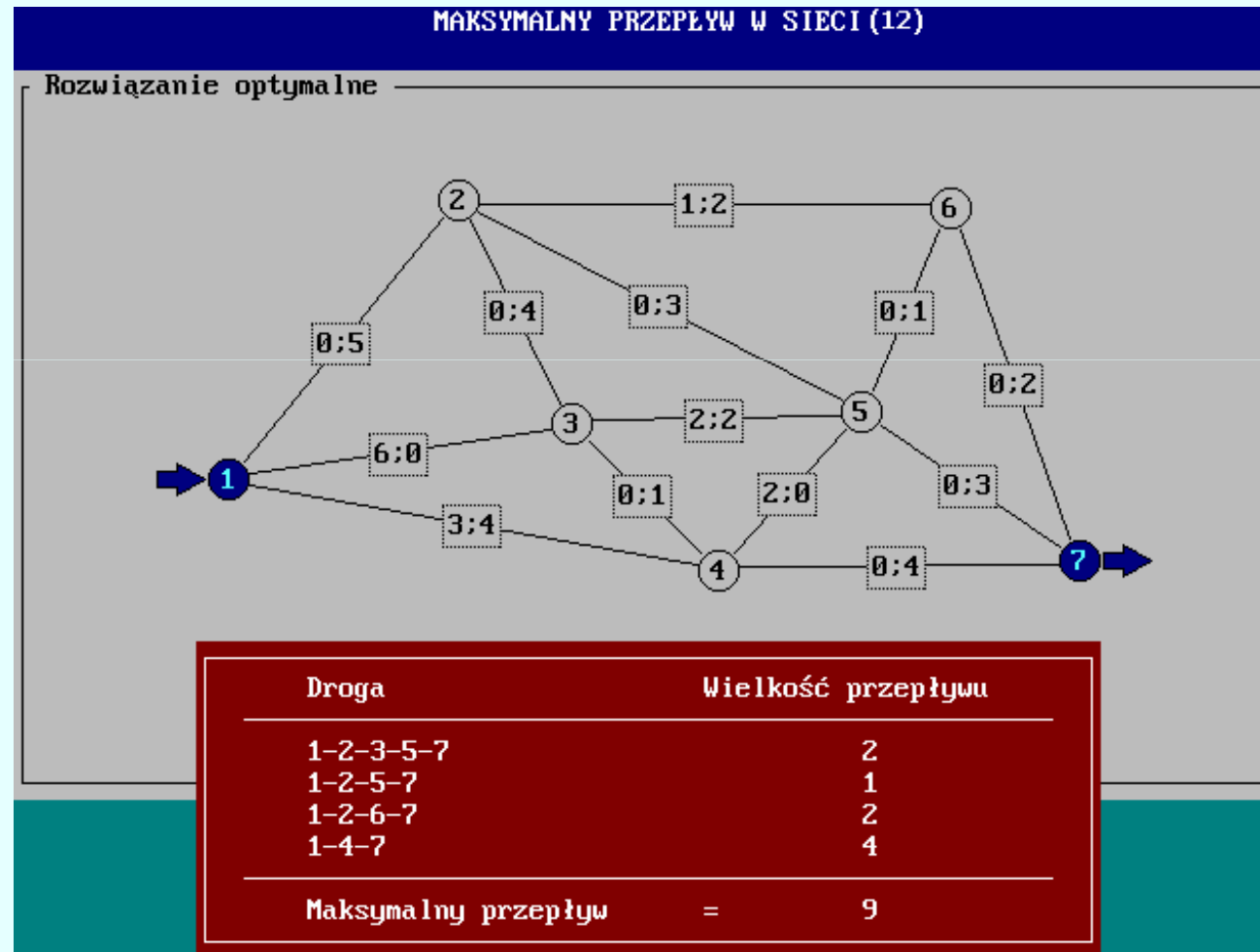
#### Iteracja 4



## 8.4. Maksymalny przepływ w sieci

### 8.4.2. Kolejne iteracje (9/9)

#### Rozwiązanie optymalne



## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.1. Dynamiczny problem wielkości zamówienia (1/4)

#### Przykład 8.4

Koszt zakupu = 120,

Jednostkowy koszt utrzymania zapasu niezależny od długości okresu magazynowania = 3,

Popyt:

miesiąc	popyt
1	35
2	70
3	17
4	52
5	42
6	20

Określić optymalny plan zaspokojenia popytu, uwzględniający koszty zakupu i koszty magazynowania

## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.1. Dynamiczny problem wielkości zamówienia (2/4)

#### *Koszty zakupu i utrzymania zapasu*

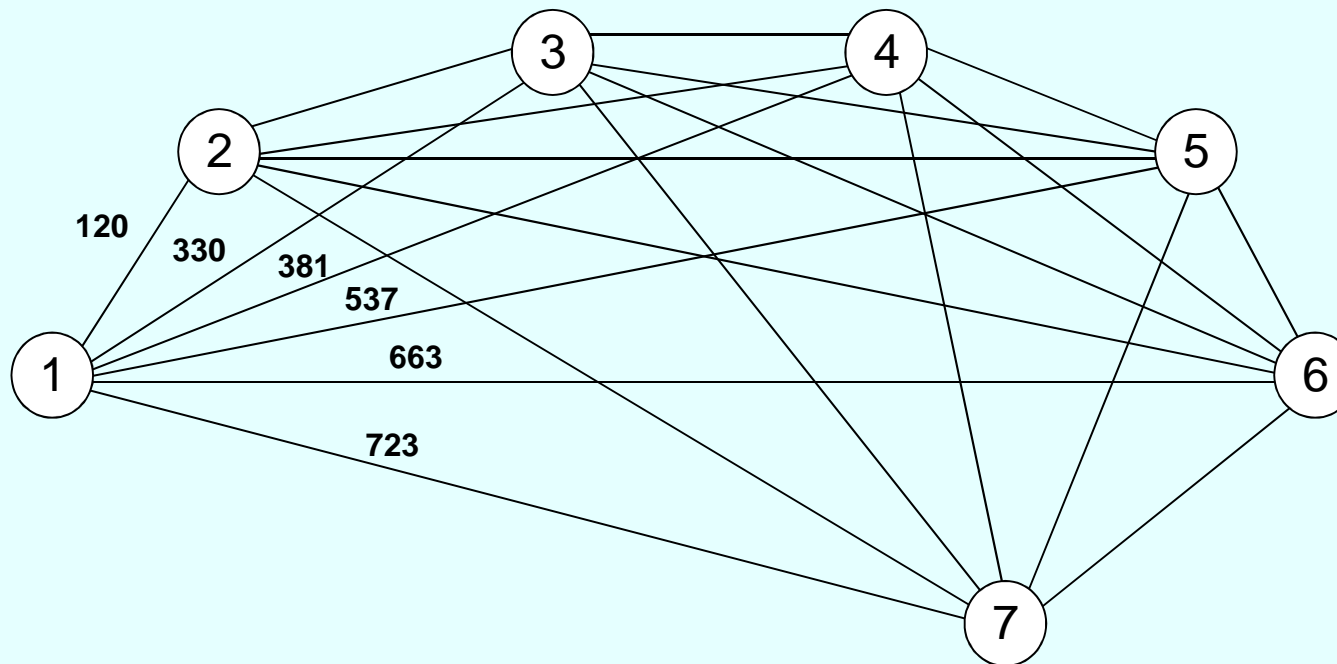
Krawędź	Koszt
1-2	120
1-3	$120 + 70 \cdot 3 = 330$
1-4	$120 + 87 \cdot 3 = 381$
1-5	$120 + 139 \cdot 3 = 537$
1-6	$120 + 181 \cdot 3 = 663$
1-7	$120 + 201 \cdot 3 = 723$
2-3	120
2-4	$120 + 17 \cdot 3 = 171$
2-5	$120 + 69 \cdot 3 = 327$
2-6	$120 + 111 \cdot 3 = 453$
2-7	$120 + 131 \cdot 3 = 513$

Krawędź	Koszt
3-4	120
3-5	$120 + 52 \cdot 3 = 276$
3-6	$120 + 94 \cdot 3 = 402$
3-7	$120 + 114 \cdot 3 = 462$
4-5	120
4-6	$120 + 42 \cdot 3 = 246$
4-7	$120 + 62 \cdot 3 = 306$
5-6	120
5-7	$120 + 20 \cdot 3 = 180$
6-7	120

## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.1. Dynamiczny problem wielkości zamówienia (3/4)

*Zastosowanie algorytmu najkrótszych dróg w sieci*



## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.1. Dynamiczny problem wielkości zamówienia (4/4)

#### *Rozwiązanie i interpretacja*

#### **Rozwiązanie optymalne**

Najkrótsza droga od wierzchołka początkowego do wierzchołka końcowego przechodzi przez wierzchołki 1-2-4-5-7. Koszt przejścia tej drogi wynosi 591.

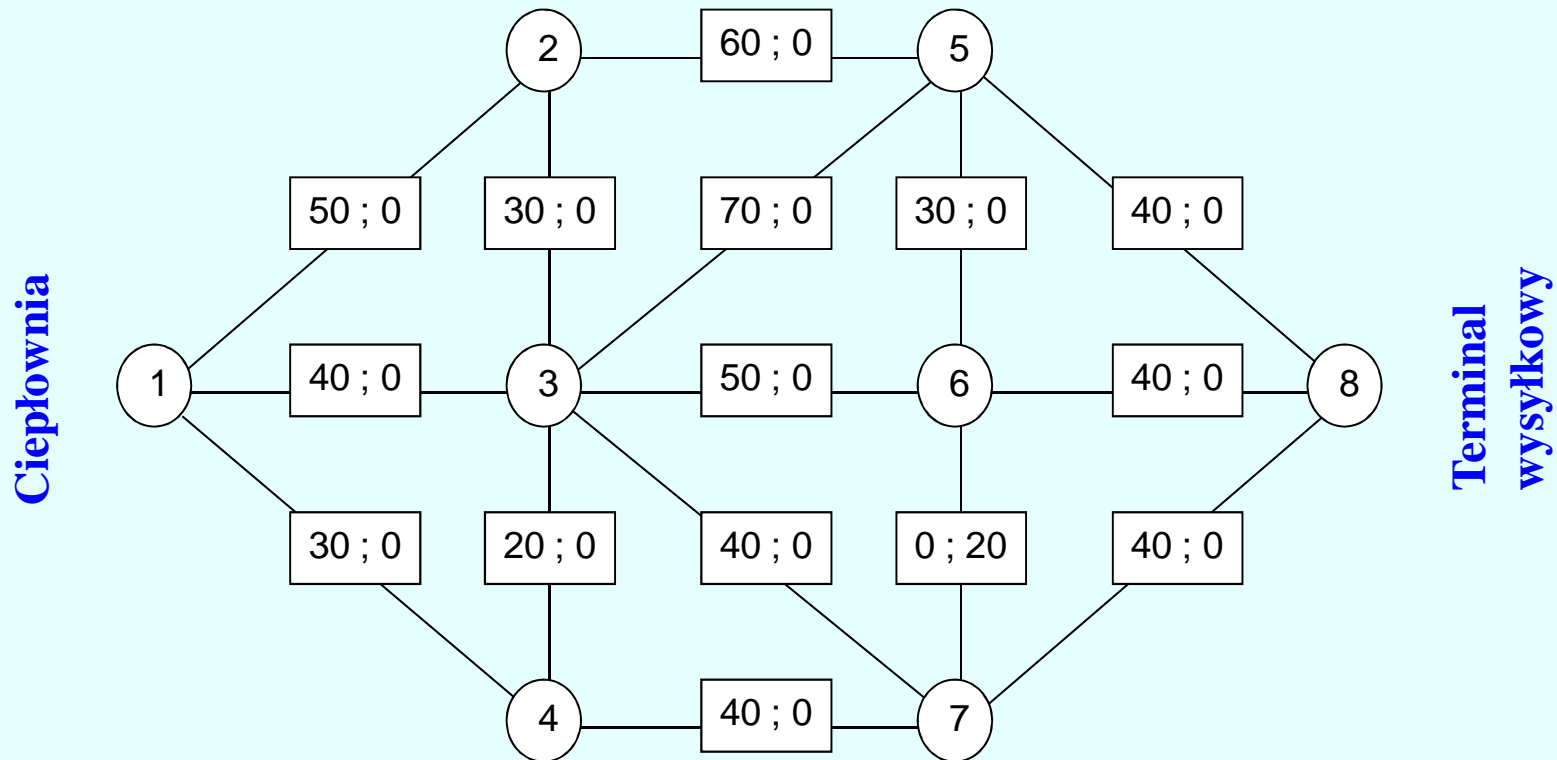
#### **Interpretacja rozwiązania**

Optymalny plan zakupów przewiduje dokonanie zakupu 35 jednostek w pierwszym miesiącu (krawędź 1-2: 35), 87 jednostek na dwa kolejne miesiące (krawędź 2-4: 70 + 17), następnie 52 jednostek (krawędź 4-5: 52) i 62 jednostek na ostatnie dwa miesiące (krawędź 5-7: 42 + 20).

## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.2. Optymalizacja transportu gorącej wody z ciepłowni do terminala wysyłkowego (1/4)

#### Przykład 8.5



Znaleźć wielkość maksymalnego przepływu z ciepłowni do terminala oraz określić, jakie ilości ciepłej wody będą płynęły przez poszczególne odcinki.



## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.2. Optymalizacja transportu gorącej wody z ciepłowni do terminala wysyłkowego (2/4)

#### Rozwiązanie optymalne

droga	wielkość przepływu
1-2-3-4-7-8	20
1-2-3-5-8	10
1-2-5-8	20
1-3-2-5-8	10
1-3-2-5-6-8	20
1-3-5-6-8	10
1-4-3-6-8	10
1-4-3-7-8	10
1-4-7-8	10

Maksymalny przepływ = 120

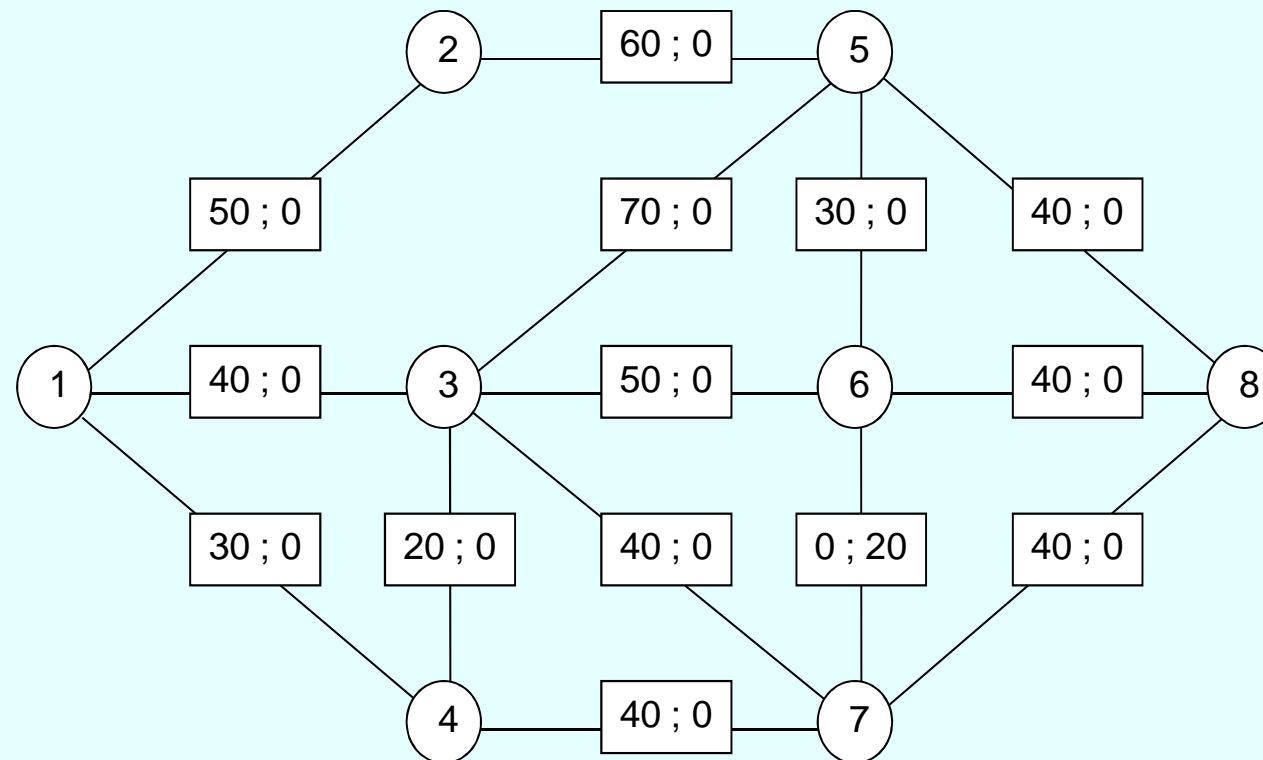
odcinek	przepływ
1-2	50
1-3	40
1-4	30
2-3	30
2-5	50
3-2	30
3-5	20
3-6	10
3-7	10
4-3	20
4-7	30
5-8	40
6-8	40
7-8	40

## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.2. Optymalizacja transportu gorącej wody z ciepłowni do terminala wysyłkowego (3/4)

#### *Modyfikacja sieci*

Awaria odcinka 2 - 3



## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.2. Optymalizacja transportu gorącej wody z ciepłowni do terminala wysyłkowego (4/4)

#### *Rozwiązanie optymalne zadania zmodyfikowanego*

<b>droga</b>	<b>wielkość przepływu</b>
1-2-5-8	40
1-2-5-6-8	10
1-3-4-7-8	20
1-3-5-6-8	20
1-4-3-6-8	10
1-4-3-7-8	10
1-4-7-8	10

Maksymalny przepływ = 120

## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.3. Optymalizacja przebiegu linii światłowodowej (1/3)

#### Przykład 8.6

Koszty połączeń		1	2	3	4	5	6	7	8	9	10
		Rektorat	Budynek A	Budynek B	Budynek C	Budynek D	Budynek E	Budynek F	Biblioteka	Akademik 1	Akademik 2
1	Rektorat		8	7	9	-	18	-	-	-	-
2	Budynek A	8		7	8	-	-	-	-	5	-
3	Budynek B	7	7		8	7	12	10	16	-	-
4	Budynek C	9	8	8		18	-	-	-	9	-
5	Budynek D	-	-	7	18		15	9	-	15	-
6	Budynek E	18	-	12	-	15		6	14	-	-
7	Budynek F	-	-	10	-	9	6		12	14	20
8	Biblioteka	-	-	16	-	-	14	12		5	-
9	Akademik 1	-	5	-	9	15	-	14	5		5
10	Akademik 2	-	-	-	-	-	-	20	-	5	

Zminimalizować koszty budowy sieci umożliwiającej przesyłanie informacji między dowolnymi budynkami uczelni.

## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.3. Optymalizacja przebiegu linii światłowodowej (2/3)

#### *Konstrukcja sieci*

krawędź	koszt
1-2	8
1-3	7
1-4	9
1-6	18
2-3	7
2-4	8
2-9	5
3-4	8
3-5	7
3-6	12

krawędź	koszt
4-5	18
4-9	9
5-6	15
5-7	9
5-9	15
6-7	6
6-8	14
7-8	12
7-9	14
7-10	20
8-9	5
9-10	5

## 8.5. Przykłady wykorzystania programowania sieciowego

### 8.5.3. Optymalizacja przebiegu linii światłowodowej (3/3)

*Rozwiązanie optymalne*

<b>krawędź</b>	<b>koszt</b>
1-2	7
2-3	7
2-4	8
2-9	5
3-5	7
5-7	9
6-7	6
8-9	5
9-10	5
suma	<b>59</b>

## Pora na relaks

