INFORMATYKA

WYBRANE ALGORYTMY OPTYMALIZACYJNE

KRYPTOLOGIA

http://www.infoceram.agh.edu.pl

Klasy metod algorytmicznych

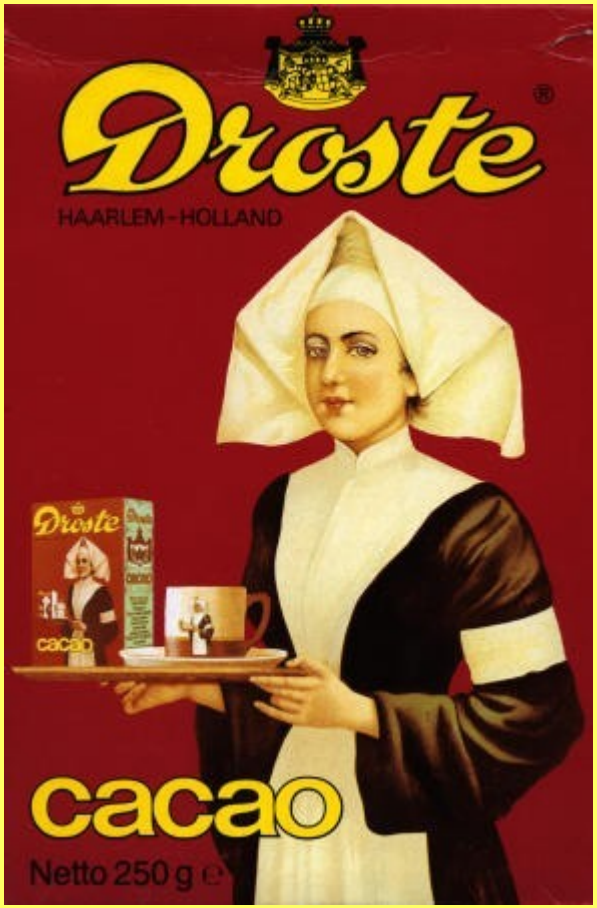
Metoda TOP-DOWN (zstępująca, analityczna)	**Metoda BOTTOM-UP (wstępująca, syntetyczna)**
• Problem jest dzielony na podproblemy, • Podproblemy są rozwiązywane, a rezultaty zapamiętywane, jeżeli będą użyte później; • Używana jest rekurencja i zapamiętywanie.	• Wszystkie (elementarne) podproblemy, które mogą być potrzebne, są rozwiązywane najpierw, • Ich rezultaty są następnie używane do rozwiązywania większych podproblemów.

Rekurencja

Rekurencja (rekursja) to odwoływanie się funkcji (podprogramu, programu) do samej siebie

Efekt Droste

Efekt Droste to termin określający specjalny rodzaj rekurencyjnego obrazu. Obraz ukazujący efekt Droste zawiera mniejszą wersję samego siebie. Ta mniejsza wersja zawiera jeszcze mniejszą wersję w tym samym miejscu i tak dalej.



Silnia

```
Function silnia(x As Double) As Double
    If x = 1 Then
        silnia = 1
    Else
        silnia = x * silnia(x - 1)
    End If
End Function
```

Podstawowe metody rozwiązywania algorytmów

- Metoda „Dziel i zwyciężaj” (divide and conquer)
- Metoda „zachłanna” (greedy method)
- Programowanie dynamiczne

Dziel i zwyciężaj – jedna z głównych metod projektowania algorytmów w informatyce, prowadząca do bardzo efektywnych rozwiązań. Nazwa pochodzi od łacińskiej sentencji dziel i rządź (łac. *divide et impera*). W strategii tej problem dzieli się rekurencyjnie na dwa lub więcej mniejszych podproblemów tego samego (lub podobnego) typu tak długo, aż fragmenty staną się wystarczająco proste do bezpośredniego rozwiązania. Z kolei rozwiązania otrzymane dla podproblemów scala się, uzyskując rozwiązanie całego zadania.

Algorytm zachłanny – algorytm, który w celu wyznaczenia rozwiązania w każdym kroku dokonuje zachłannego, tj. najlepiej rokującego w danym momencie wyboru rozwiązania częściowego. Innymi słowy algorytm zachłanny nie dokonuje oceny czy w kolejnych krokach jest sens wykonywać dane działanie, dokonuje decyzji **lokalnie optymalnej**, dokonuje on wyboru wydającego się w danej chwili najlepszym, kontynuując rozwiązanie podproblemu wynikającego z podjętej decyzji

Programowanie dynamiczne – opiera się na podziale rozwiązywanego problemu na podproblemy względem kilku parametrów. W odróżnieniu od techniki dziel i zwyciężaj podproblemy w programowaniu dynamicznym nie są rozłączne.

Podstawowe metody rozwiązywania algorytmów

Algorytmy optymalizacyjne

- Metoda „zachłanna” (greedy method)
- Programowanie dynamiczne

Algorytmy zachłanne

- Własność zachłannego wyboru
  - Za pomocą lokalnie optymalnych (zachłannych) wyborów można uzyskać optymalne rozwiązanie całego zadania.
  - Optymalny wybór nie zależy od wyborów kolejnych
- Metoda zachłanna
  - Dokonuje decyzji lokalnie optymalnej (w danej chwili najlepszej)

Programowanie dynamiczne

- W każdym kroku rozważamy wszystkie kombinacje powstałe z:
  - Dokonania konkretnego wyboru (zazwyczaj zachłannego)
  - Znalezienia optymalnych rozwiązań dla pozostałych wyborów
- Optymalny wybór dokonany przez metodę programowania dynamicznego może zależeć od poprzednich wyborów oraz od wyborów kolejnych – wtedy rozwiązanie optymalne jest modyfikowane.

Dyskretny problem plecakowy

Złodziej rabuje mieszkanie W. Skibińskiego. Znalazł on N towarów; *j*-ty przedmiot jest wart c_j oraz waży w_j. Złodziej dąży do zabrania ze sobą jak najwartościowszego łupu, przy czym nie może zabrać więcej niż B kilogramów. Nie może też zabierać ułamkowej części przedmiotów (byłoby to możliwe w ciągłym problemie plecakowym).


PLAYSTATION 3


laptop
parasol
trampki
gitara
wino
telefon
PSVita
PS3

Torba ma pojemność 15 kg

j	przedmiot	waga, w_j	wartość, c_j
1	gitara	16	500
2	PS3	7	800
3	PS Vita	2	200
4	telefon	1	500
5	laptop	8	900
6	parasol	2	100
7	trampki	4	100
8	wino	2	100

Rozwiązanie zachłanne

W wersji zachłannej algorytm aproksymacyjny sortuje elementy w kolejności malejącej porównując stosunek wartości do wagi elementu

$$h_j = \frac{c_j}{w_j}$$

i następnie wstawia je kolejno.

Rozwiązanie zachłanne

j	przedmiot	waga, w_j	wartość, c_j	h_j
1	telefon	1	500	500
2	PS3	7	800	114.3
3	laptop	8	900	112.5
4	PS Vita	2	200	100
5	wino	2	100	50
6	parasol	2	100	50
7	gitara	16	500	31,25
8	trampki	4	100	25

Rozwiązanie zachłanne

j	przedmiot	waga, w_j	wartość, c_j	h_j
1	telefon	1	500	500
2	PS3	7	800	114.3
3	laptop	8	900	112.5
4	PS Vita	2	200	100
5	wino	2	100	50
6	parasol	2	100	50
7	gitara	16	500	31,25
8	trampki	4	100	25

Waga plecaka – 14 kg

Wartość – 1700 zł

Rozwiązanie dynamiczne

Polega na rozważeniu wszystkich możliwych opcji i wybraniu najbardziej optymalnej

j	1	1 i 2	1 i 3	1 i 4	1 i 5	1 i 6	…
waga	1	8	9	3	3	3	…
wartość	500	1300	1400	700	600	600	…

Rozwiązanie optymalne

j	przedmiot	waga, w_j	wartość, c_j	h_j
1	telefon	1	500	500
2	PS3	7	800	114.3
3	laptop	8	900	112.5
4	PS Vita	2	200	100
5	wino	2	100	50
6	parasol	2	100	50
7	gitara	16	500	31,25
8	trampki	4	100	25

Waga plecaka – 15 kg

Wartość – 1800 zł

Problem wydawania reszty

Problem polegający na wybraniu z danego zbioru monet o określonych nominałach takiej konfiguracji, by wydać żądaną kwotę przy użyciu minimalnej liczby monet.

Przykładowe zadanie

- Dane są trzy nominały – 1 zł, 2 zł i 5 zł.
- Ile minimalnie monet potrzeba, żeby wydać 13 zł?
- zał. Posiadamy nieskończony zbiór monet

Algorytm zachłanny

k – żądana kwota = 13 zł

n – największy dostępny (mniejszy od żądanej kwoty) nominał

x – liczba potrzebnych monet

k	13	8	3	1
n	5	5	2	1
x	4	3	2	1

Algorytm zachłanny – zbiór nominałów 2 zł, 5 zł

k – żądana kwota = 6 zł

n – największy dostępny (mniejszy od żądanej kwoty) nominał

x – liczba potrzebnych monet

k	6	1
n	5	?????
x	1	?????

BRAK ROZWIĄZANIA?

Programowanie dynamiczne

Dzięki wykorzystaniu programowania dynamicznego jest możliwe znalezienie bezbłędnego rozwiązania dla tego problemu przy dowolnym zbiorze nominałów i dowolnej kwocie. Algorytm polega na przetwarzaniu kolejnych nominałów i obliczeniu minimalnej liczby potrzebnych monet dla wydania kwot od 0 do k. Przy analizie kolejnego nominału wykorzystywane są informacje pozyskane w czasie wcześniejszych analiz.

Programowanie dynamiczne

k – żądana kwota = 13 zł

n – dostępne nominały

n	**T**													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
-	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1, 2	0	1	1	2	2	3	3	4	4	5	5	6	6	7
1,2,5	0	1	1	2	2	1	2	2	3	3	2	3	3	4

Czy w takim razie algorytmy zachłanne mają sens?

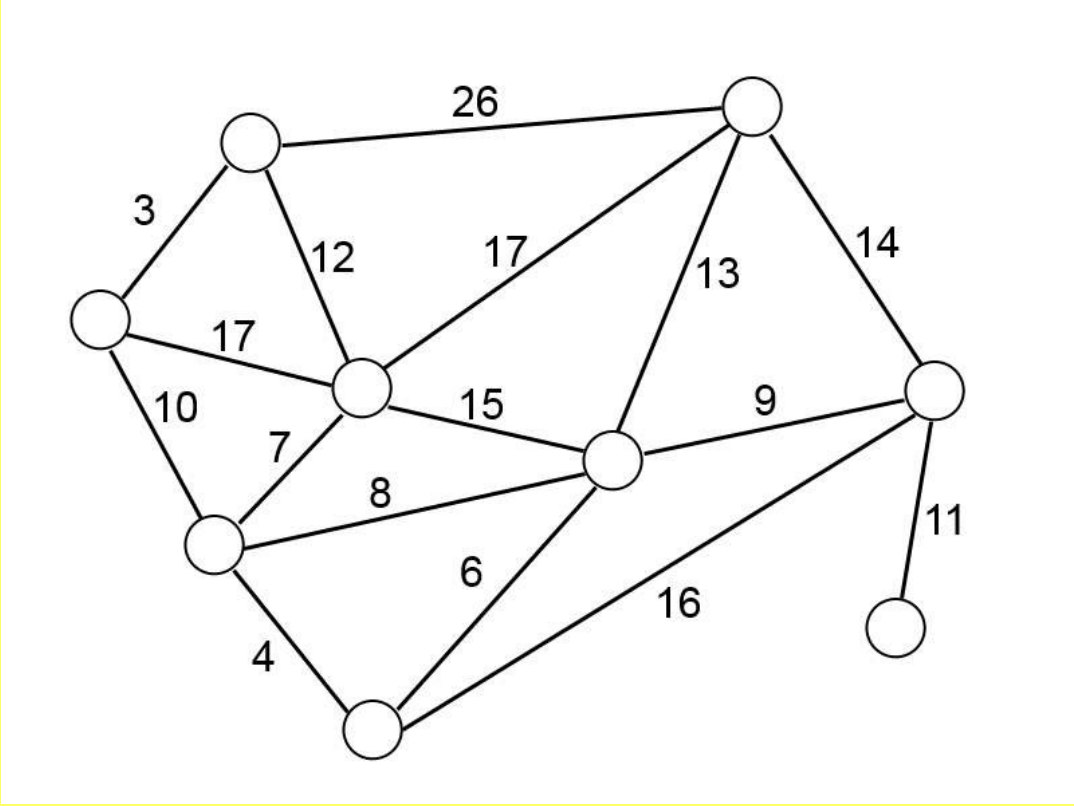
Metoda zachłanna daje optymalne rozwiązanie w niektórych algorytmach:

- Algorytm Prima
- Algorytm Kruskala
- Ciągły problem plecakowy

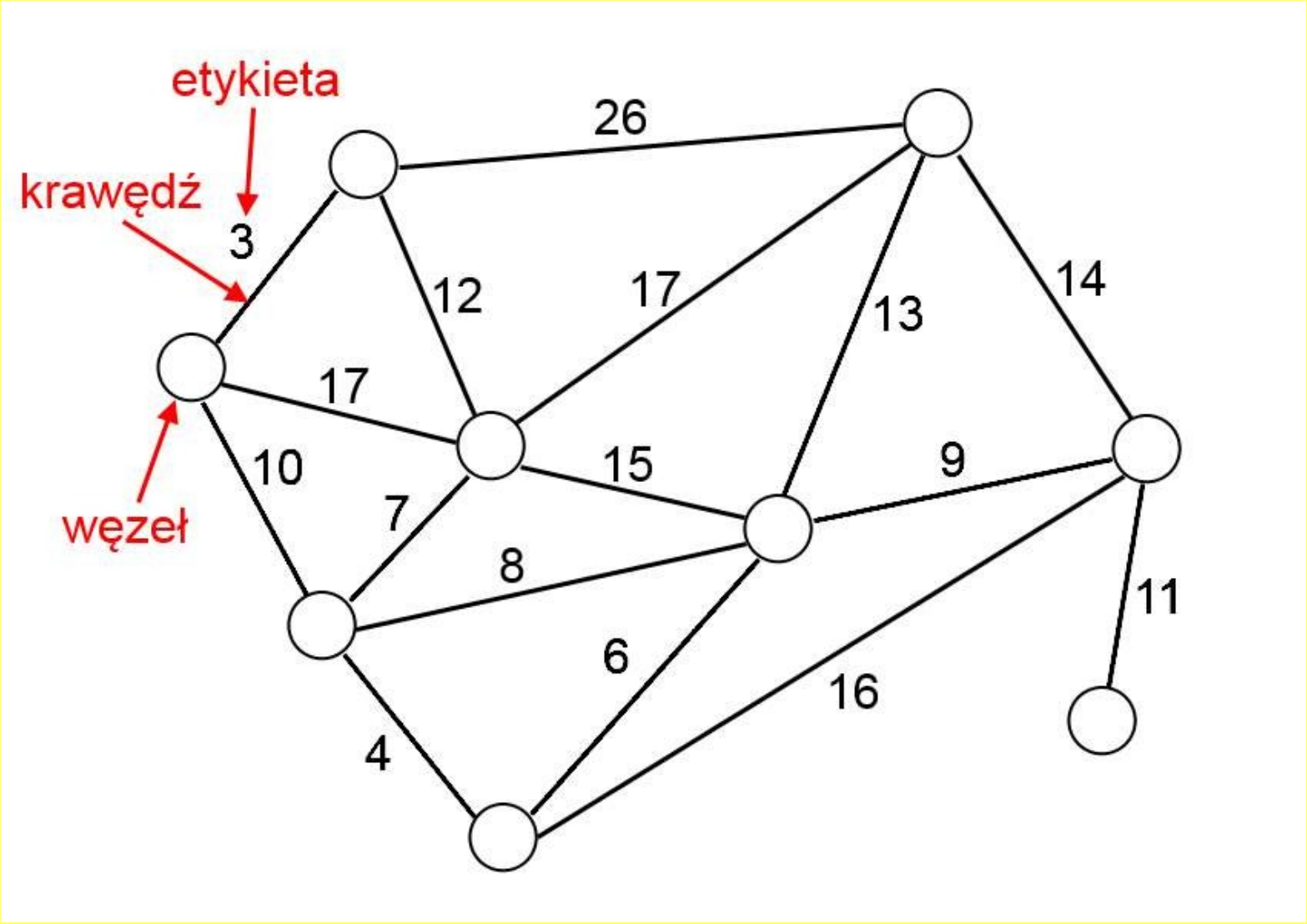
Daje wynik w dużo krótszym czasie niż programowanie dynamiczne.

Graf

Zbiór *wierzchołków*, które **mogą** być połączone *krawędziami*, w taki sposób, że każda krawędź kończy się i zaczyna w którymś z wierzchołków

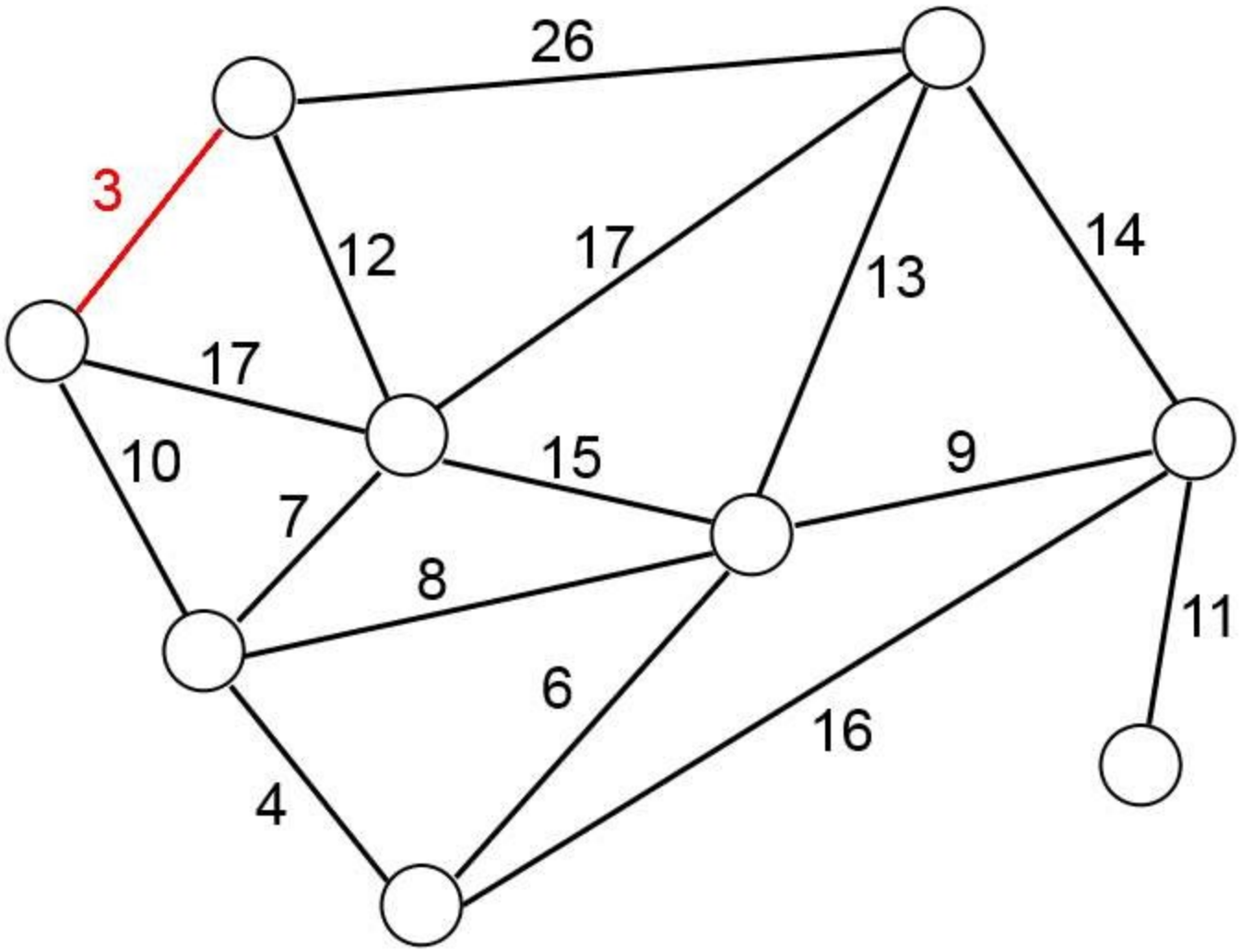


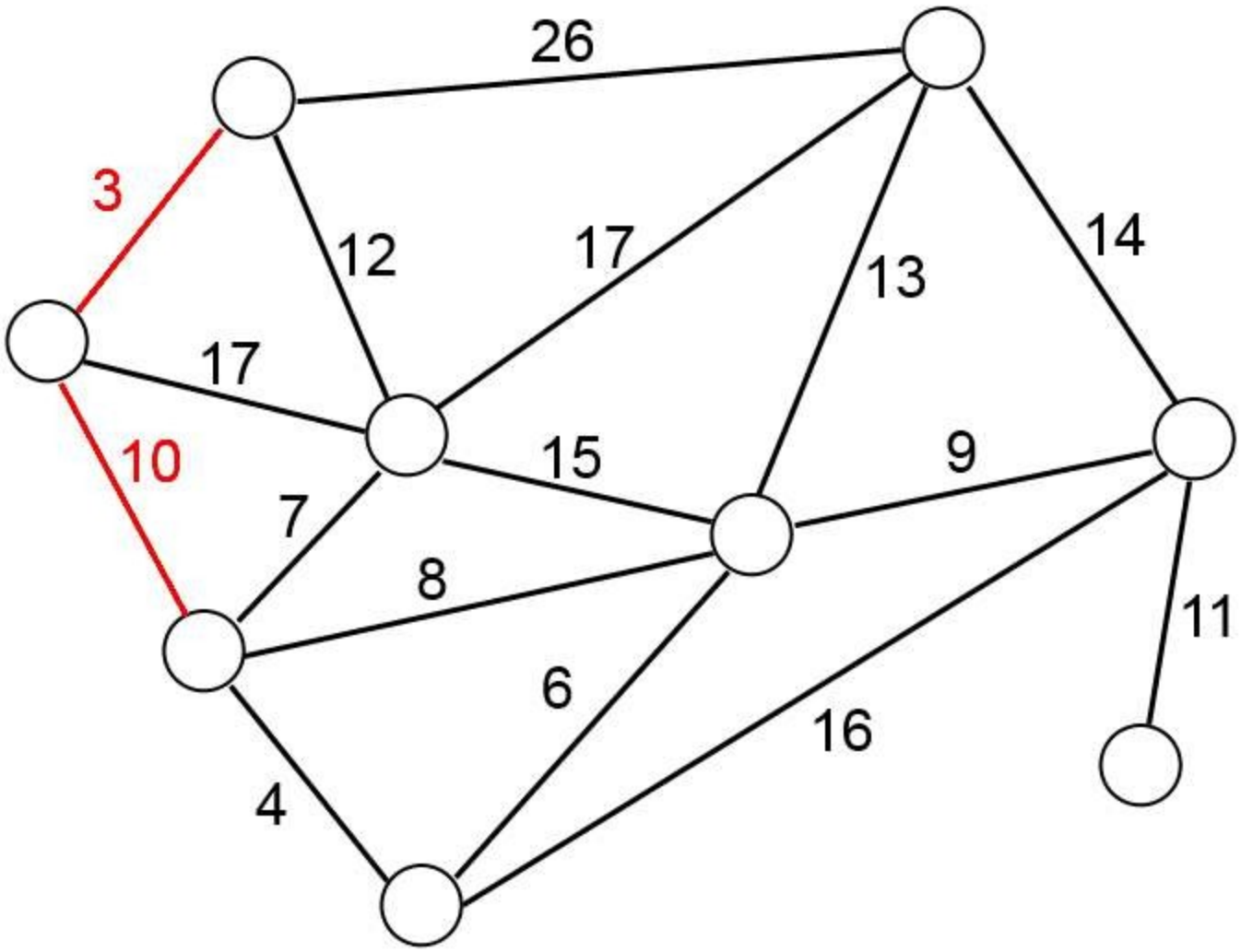
Graf

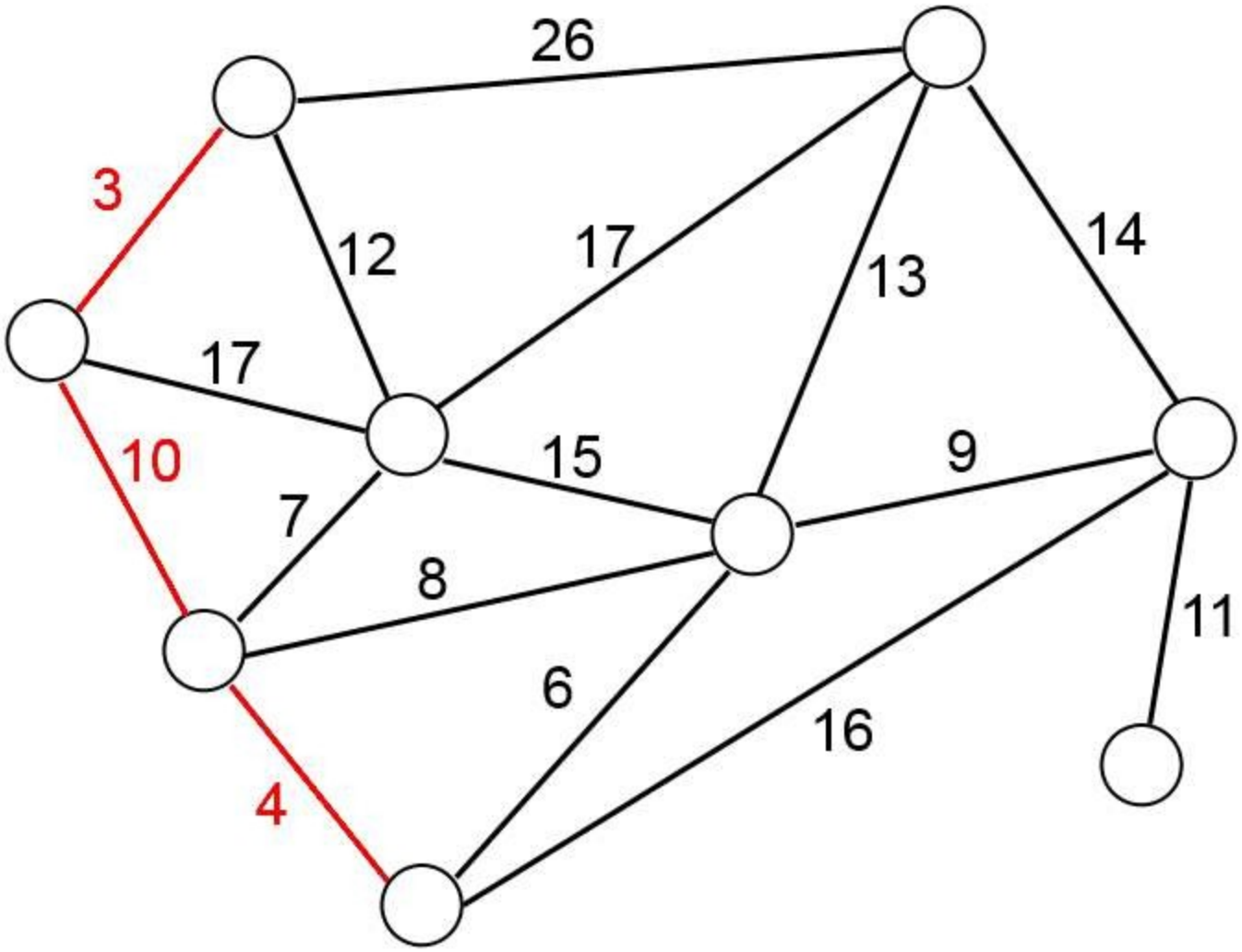


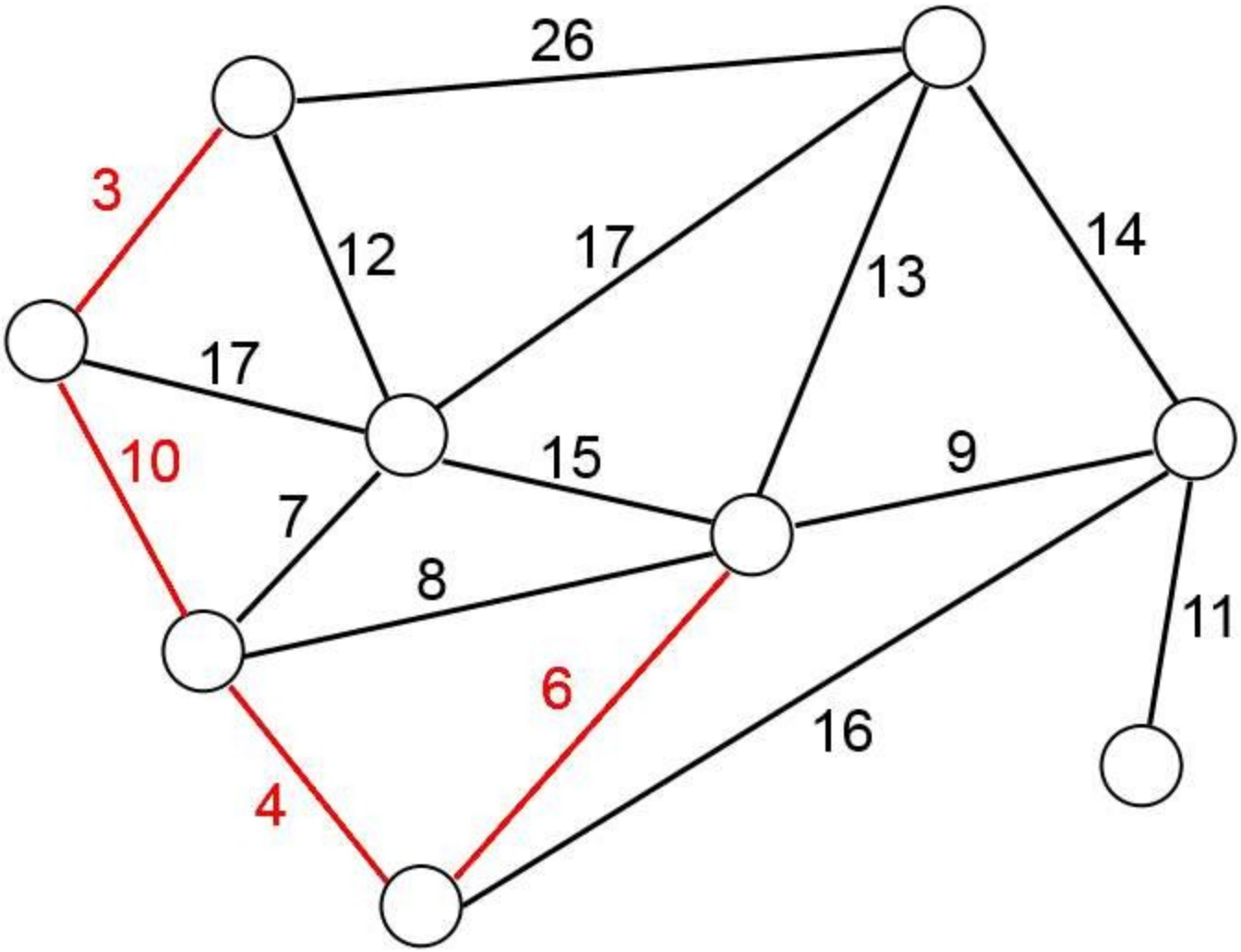
Algorytm Prima

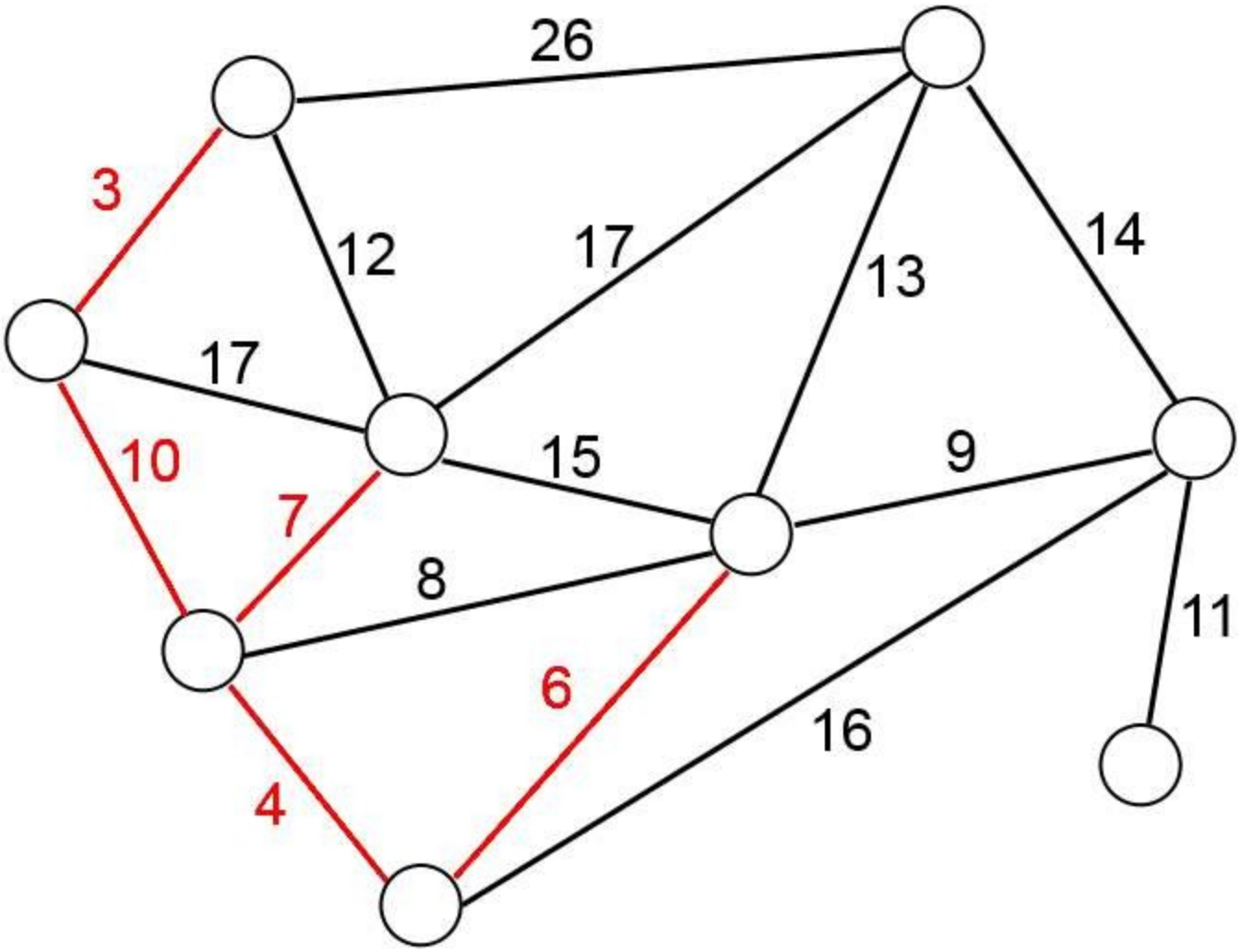
- Zbuduj drzewo zdegenerowane, składające się z najtańszej krawędzi grafu;
- W każdym kolejnym kroku dodaj do już istniejącego drzewa najtańszą krawędź z krawędzi dotąd nie wziętych pod uwagę;
- Dodanie nowej krawędzi nie może prowadzić do powstania cyklu, w takim przypadku przejdź do nowej krawędzi w porządku rosnących etykiet

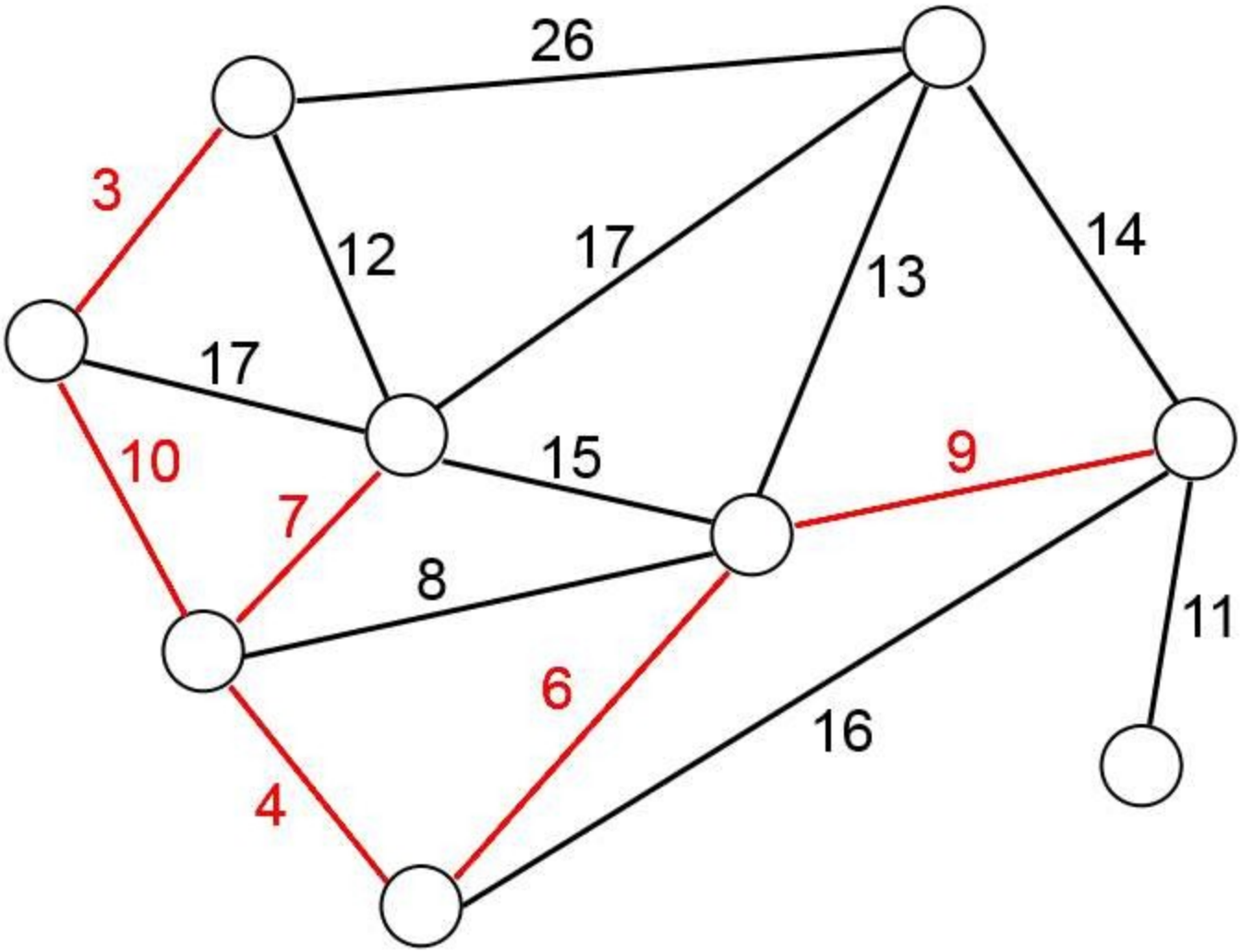

26
3
12
17
13
14
17
10
15
9
7
8
11
6
16
4

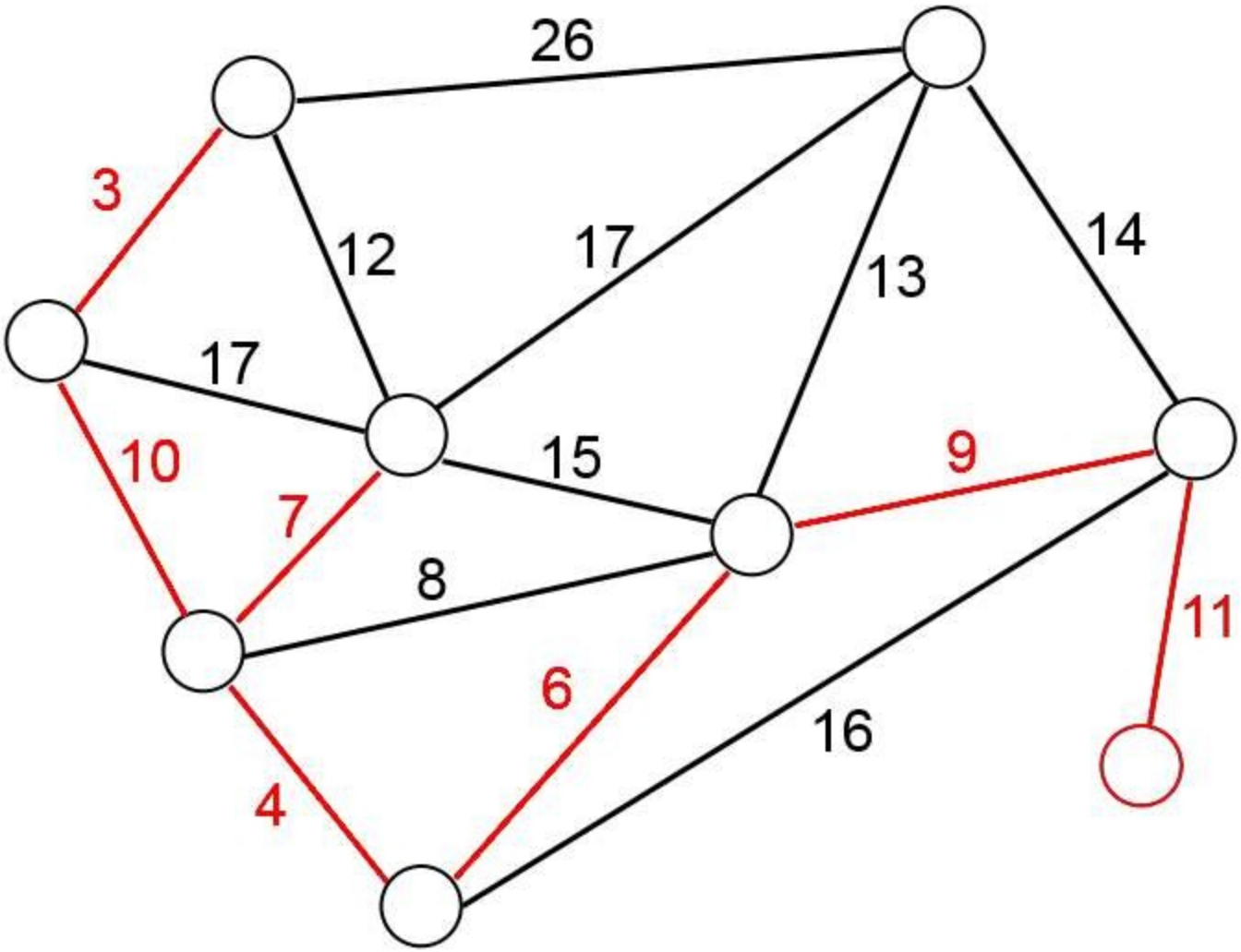

26
3
12
17
13
14
17
10
15
9
7
8
11
6
16
4

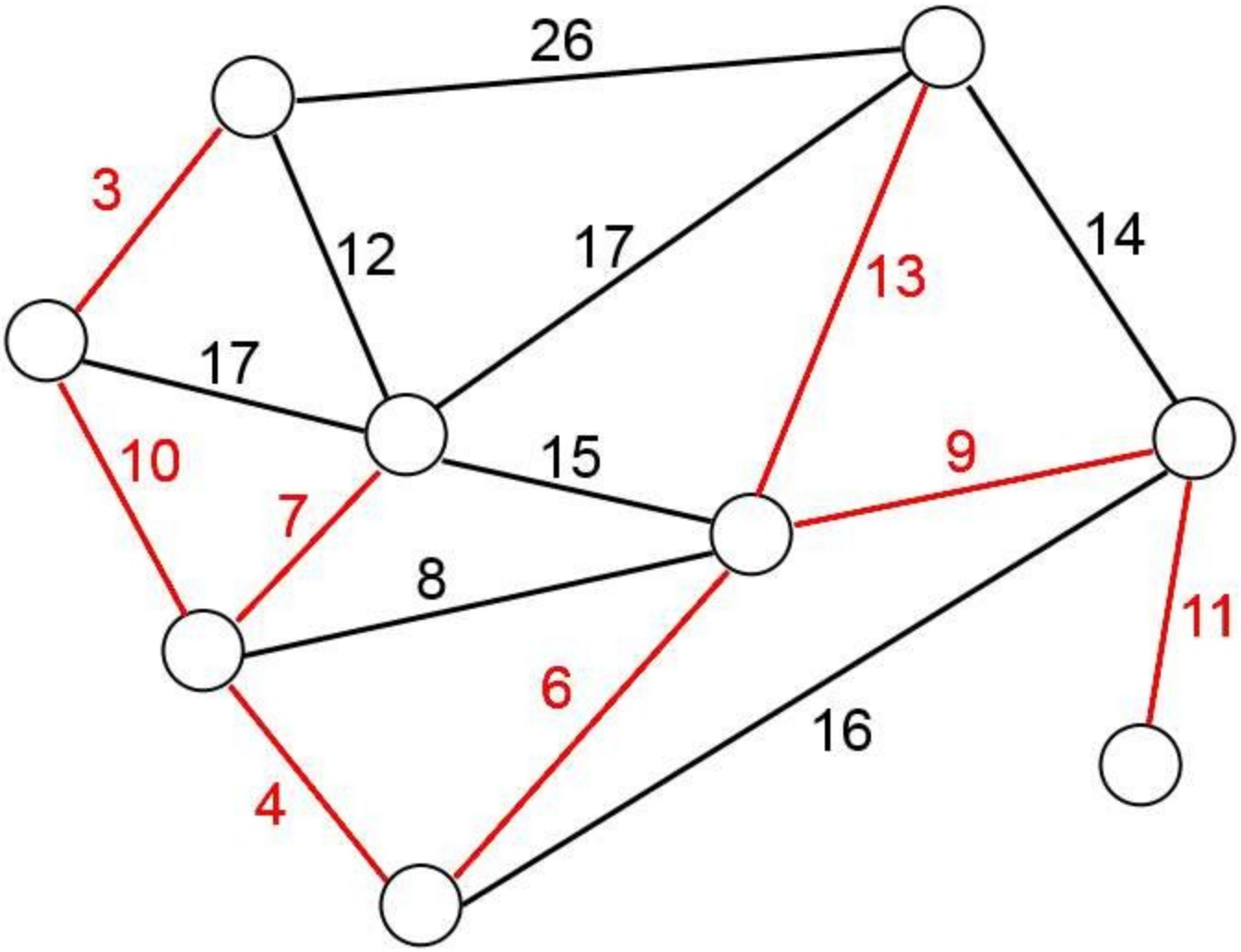

26
3
12
17
13
14
17
10
15
9
7
8
11
6
16
4

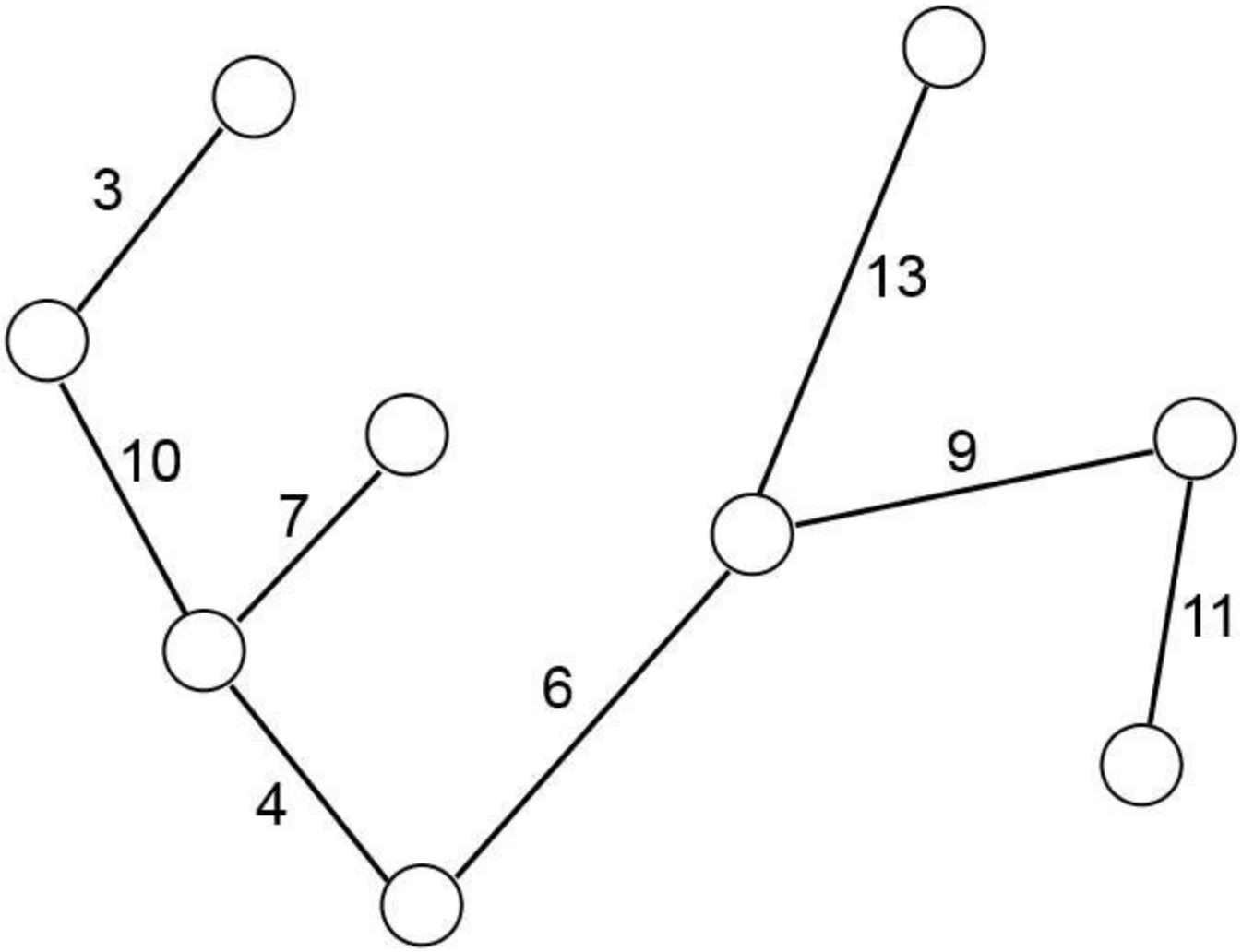

26
3
12
17
13
14
17
10
15
9
7
8
11
6
16
4


26
3
12
17
13
14
17
10
15
9
7
8
11
6
16
4


26
3
12
17
13
14
17
10
15
9
7
8
11
6
16
4


26
3
12
17
13
14
17
10
15
9
7
8
11
6
16
4


26
3
12
17
13
14
17
10
15
9
7
8
11
6
16
4


3
10
7
13
9
11
6
4

Programowanie (planowanie) dynamiczne

Algorytm znajdowania najkrótszej ścieżki w grafie

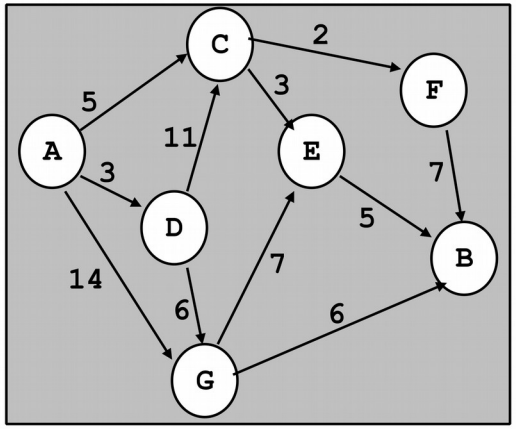
Jeśli węzłami są C_1, .., C_N i ścieżka ma się rozpocząć w C_1 i skończyć w C_N, to algorytm wymaga obliczenia optymalnej ścieżki częściowej $L(C_I)$, przedstawiającej najkrótszą ścieżkę z C_I do C_N, dla każdego I=1..N:

$L(C_I)$ == min(odległość(C_I, C_K) + $L(C_K)$)

przy czym C_K to wszystkie węzły, do których prowadzą bezpośrednio krawędzie z C_I.

Z założenia graf jest acykliczny, możliwe jest więc obliczenie wszystkich $L(C_I)$, posuwając się z B do tyłu:

1. Obliczamy L(F) = 7, L(E) = 5, L(G) = 6
2. Obliczamy L(C) = min(2 + L(F), 3 + L(E)) =
   3 + L(E) = 8
3. Obliczamy L(D) = min(11 + L(C), 6 + L(G)) =
   6 + L(G) = 12
4. Obliczamy L(A) = min(5 +L(C), 3 + L(D), 14 + L(G)) =
   5 + L(C) = 13

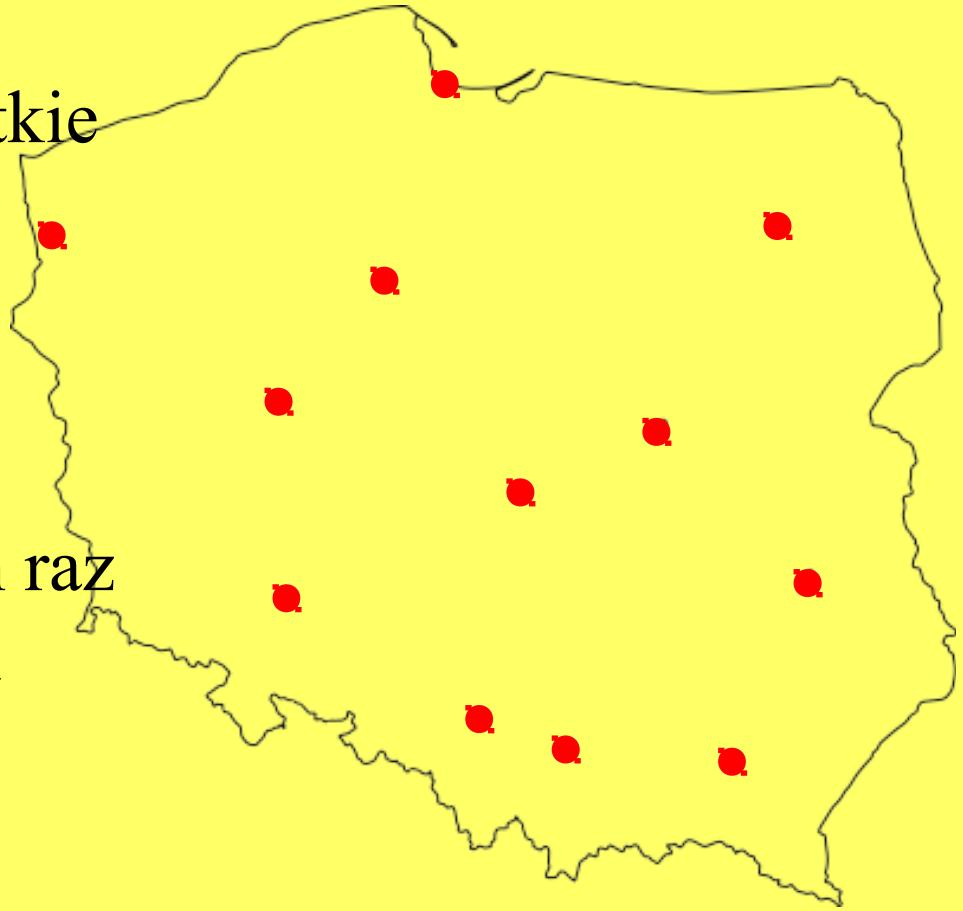


Optymalna ścieżka z A do B to: A → C → E → B

Problem komiwojażera

Polega na znalezieniu najkrótszej drogi łączącej wszystkie miasta zaczynającej się i kończącej się w określonym punkcie.

- Należy odwiedzić wszystkie miasta
- Należy wrócić do miasta początkowego
- Każde miasto można odwiedzić TYLKO jeden raz
- Kolejność odwiedzanych miast jest dowolna



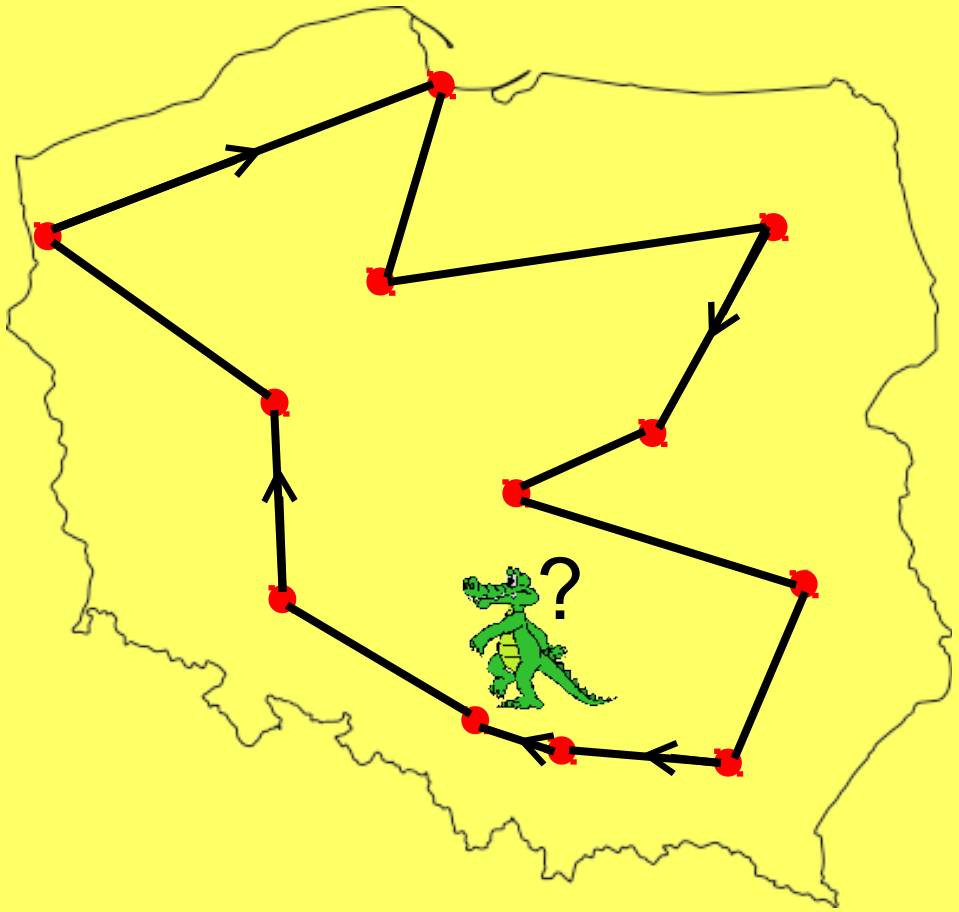
Problem komiwojażera

Liczba kombinacji:

$$R = \frac{(n-1)!}{2}$$

Dla n = 10 miast istnieje 181 440 kombinacji.

Dla n=11 miast istnieje już 1 814 400 możliwości



Problem komiwojażera

Znalezienie optymalnej trasy metodami wymaga porównania wszystkich możliwości i może zająć bardzo dużo czasu!

Z tego powodu stosuje się algorytmy genetyczne które zwracają rozwiązanie **suboptymalne!**

Czyli możliwie dobre (zadowalające) rozwiązanie osiągalne w krótkim czasie.

KRYPTOLOGIA

Systemy kryptograficzne

Kryptologia (z gr. κρυπτός – *kryptos* – "ukryty" i λόγος – *logos* – "słowo") – nauka o przekazywaniu informacji w sposób zabezpieczony przed niepowołanym dostępem.

Najwcześniejsze formy utajniania pisemnych wiadomości – z uwagi na fakt, że większość ludzi i tak nie umiała czytać – wymagały niewiele więcej niż ówczesnego odpowiednika pióra i papieru.

Systemy kryptograficzne

Zwiększenie się umiejętności czytania i pisania, szczególnie u przeciwnika, przyczyniło się do powstania rzeczywistej kryptografii.

Szyfry antyczne dzieli się na dwie główne grupy:

szyfry przestawieniowe, za pomocą których zmieniano kolejność liter w wiadomości (przykład najprostszego przestawienia – "pomóż mi" staje się "opómż im") oraz

szyfry podstawieniowe, które polegały na zastępowaniu pojedynczych liter lub ich grup, odpowiednio: innymi literami lub ich grupami (np. "natychmiastowy wylot" staje się "obuzdinjvbtupxz xzmpu" w najprostszym podstawieniu za daną literę – następnej litery alfabetu łacińskiego).

Systemy kryptograficzne

W prostych wersjach obydwa szyfry oferują niewielki stopień utajnienia przed przeciwnikiem. Jednym z najwcześniejszych szyfrów podstawieniowych był szyfr Cezara, w którym każda litera tekstu jawnego zastępowana była literą oddaloną o pewną ustaloną liczbę pozycji w alfabecie. Szyfr ten został nazwany na cześć Juliusza Cezara, który używał go (z przesunięciem o 3) do komunikacji ze swoimi generałami podczas kampanii wojskowych.

Po odkryciu metod kryptoanalizy statystycznej przez arabskiego uczonego Al-Kindiego w IX wieku n.e. stało się możliwe, z mniejszymi lub większymi trudnościami, złamanie prawie każdego z takich szyfrów

Systemy kryptograficzne

Sytuacja bezbronności szyfrów wobec kryptoanalizy panowała do momentu opracowania szyfrów polialfabetycznych około roku 1467. Pomysł polegał na użyciu różnych szyfrów (np. szyfrów podstawieniowych) dla różnych części wiadomości – często innego szyfru dla każdej z osobna litery tekstu jawnego.