

## Algorytmy grafowe 07: Generowanie drzew rozpiętych.

### A Zadania na rozgrzewkę przed egzaminem/kolokwium zaliczeniowym - nie obowiązkowe

**Zadanie A.1.** (Dla cierpliwych) Zanalizuj działanie algorytmu znajdujące wszystkie drzewa rozpięte w cyklu długości cztery z listą krawędzi jak poniżej:

$$(1, 2) (1, 4) (2, 3) (3, 4).$$

Wypisz zmiany etykiet, kolejne rozpatrywane krawędzie, co z nimi robisz (dodana/usunięta/nic), jakie kraw. są aktualnie dodane.

kraw.	działanie	1	2	3	4	aktualnie dodane

**Zadanie A.2.** Na pewnym etapie działania algorytmu znajdujące wszystkie drzewa rozpięte grafu na czternastu wierzchołkach etykiety wierzchołków wynosiły:

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$v_9$	$v_{10}$	$v_{11}$	$v_{12}$	$v_{13}$	$v_{14}$
$r(\cdot)$	1	1	1	1	5	6	1	1	1	6	5	5	5	5
$p(\cdot)$	—	3	1	3	—	—	8	1	8	6	5	11	5	13

Narysuj las wyznaczony przez te etykiety. Jakie będą się zmieniały etykiety wierzchołków (i krawędzie lasu) przy badaniu czterech następnych na liście krawędzi (zakończ działanie algorytmu dopiero w momencie gdy był(a)byś zmuszon(a)y do rozpatrywania następnej, która nie jest już podana, krawędzi :

$$(v_3, v_{12}), (v_7, v_{14}), (v_6, v_7), (v_7, v_{10})?$$

Wypisz też kolejne rozpatrywane krawędzie i co z nimi robisz (dodana/usunięta/nic).

Przykładowe rozwiązania na końcu pliku.

### B Program do napisania

Proszę o przesłanie

- do nocy ze środy na czwartek (**22/23 kwietnia**);
- w mailu o tytule **AGR04** (WAŻNE: Nie będę czytała tych maili, więc z istotnymi sprawami proszę się zgłaszać w osobnych mailach.)
- plików o zindywidualizowanej nazwie **04NazwiskoImie.py** albo **04NazwiskoImie.txt** (jeśli .py nie chce się wysłać) albo skompresowane o nazwie **04NazwiskoImie** (ale TYLKO jeśli piszą Państwo w kilku plikach) albo **04NazwiskoImieNieDziała.\*** (jeśli podjęli Państwo próbę zrobienia, ale nie działa);
- na adres: kryba@amu.edu.pl.
- **Proszę:**
  - **nazwisko pierwsze, bez polskich znaków;**
  - **nie wysyłać niekompletnych programów bez dopisku NieDziała;**
  - jak wysyłają Państwo nową, **poprawioną wersję**, to **dodać numer wersji** np. 02 na końcu nazwy.
- Proszę o wpisanie w programie '*graph07.txt*' a nie odwołania do pliku, które Państwo wykorzystywali.
- Proszę nie wysyłać mi pliku tekstowego z grafem.

**Zadanie B.1.** (Zadanie do wysłania) W pliku *graph07.txt* zapisana jest macierz wag pewnego grafu (wagi CAŁKOWITOLICZBOWE). Napisz program, który wykorzystując algorytm Prima znajduje minimalne drzewo rozpięte wczytanego grafu. W wyjściu mają się znaleźć:

- kolejno wypisane rozpatrywane wierzchołki (zaczynając działanie od wierzchołka 1) a za nimi
- kolejno wypisane stany etykiet po rozpatrzeniu wierzchołka;
- krawędzie i waga znalezionej drzewa.

**Zadanie B.2.** (Dla ciekawskich - proszę nie wysyłać) W pliku *paths.txt* zapisana jest macierz wag pewnego grafu (wagi DODATNIE CAŁKOWITOLICZBOWE). Zmodyfikuj poprzedni program tak, aby znajdował najkrótsze ścieżki z wierzchołka 1 z wykorzystaniem algorytmu Dijkstry (W zasadniczej części algorytmu wcale dużo nie trzeba zrobić. Zależy od implementacji, ale może się okazać, że tylko jedną linijkę.) W wyjściu mają się znajdować kolejno:

- kolejno wypisane rozpatrywane wierzchołki (zaczynając działanie od wierzchołka 1) a za nimi
- kolejno wypisane stany etykiet po rozpatrzeniu wierzchołka;
- wypisane znalezione najkrótsze ścieżki z ich wagami. (W tym punkcie trzeba więcej zmienić, ale można skorzystać z przerobionego kodu z algorytmu Floyd-Warshalla - odczytywanie ścieżek działa analogicznie.)

## B1

### PRZYKŁADOWE WEJŚCIE:

```
- 9 - 3 - - 6
9 - 13 - - 15 8
- 13 - 14 4 17 -
3 - 14 - - - 1
- - 4 - - 11 -
- 15 17 - 11 - -
6 8 - 1 - - -
```

### PRZYKŁADOWE WYJŚCIE:

```
Rozważany wierzchołek: 1
[None, 0] [1, 9] [None, inf] [1, 3] [None, inf] [None, inf] [1, 6]
Rozważany wierzchołek: 4
[None, 0] [1, 9] [4, 14] [1, 3] [None, inf] [None, inf] [4, 1]
Rozważany wierzchołek: 7
[None, 0] [7, 8] [4, 14] [1, 3] [None, inf] [None, inf] [4, 1]
Rozważany wierzchołek: 2
[None, 0] [7, 8] [2, 13] [1, 3] [None, inf] [2, 15] [4, 1]
Rozważany wierzchołek: 3
[None, 0] [7, 8] [2, 13] [1, 3] [3, 4] [2, 15] [4, 1]
Rozważany wierzchołek: 5
[None, 0] [7, 8] [2, 13] [1, 3] [3, 4] [5, 11] [4, 1]
Rozważany wierzchołek: 6
Krawędzie drzewa: (2, 7),(3, 2),(4, 1),(5, 3),(6, 5),(7, 4),
Waga drzewa: 40
```

## B2

### PRZYKŁADOWE WEJŚCIE:

```
- 6 - - - 8 2 -
6 - - - 2 - 3 -
- - - 4 - 5 - 3
- - 4 - - 5 - -
- 2 - - - - 6 -
8 - 5 5 - - - 9
2 3 - - 6 - - -
- - 3 - - 9 - -
```

### PRZYKŁADOWE WYJŚCIE:

```
Rozpatrywamy wierzchołek: 1
(0, None) (6, 1) (inf, None) (inf, None) (inf, None) (8, 1) (2, 1) (inf, None)
Rozpatrywamy wierzchołek: 7
(0, None) (5, 7) (inf, None) (inf, None) (8, 7) (8, 1) (2, 1) (inf, None)
Rozpatrywamy wierzchołek: 2
(0, None) (5, 7) (inf, None) (inf, None) (7, 2) (8, 1) (2, 1) (inf, None)
Rozpatrywamy wierzchołek: 5
(0, None) (5, 7) (inf, None) (inf, None) (7, 2) (8, 1) (2, 1) (inf, None)
Rozpatrywamy wierzchołek: 6
(0, None) (5, 7) (13, 6) (13, 6) (7, 2) (8, 1) (2, 1) (17, 6)
Rozpatrywamy wierzchołek: 3
(0, None) (5, 7) (13, 6) (13, 6) (7, 2) (8, 1) (2, 1) (16, 3)
Rozpatrywamy wierzchołek: 4
(0, None) (5, 7) (13, 6) (13, 6) (7, 2) (8, 1) (2, 1) (16, 3)
Najkrótsza ścieżka z 1 do:
```

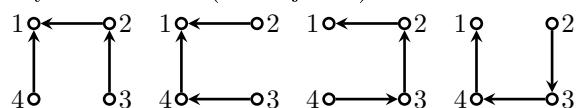
1 : 1 , dlugosc: 0  
 2 : 1 7 2 , dlugosc: 5  
 3 : 1 6 3 , dlugosc: 13  
 4 : 1 6 4 , dlugosc: 13  
 5 : 1 7 2 5 , dlugosc: 7  
 6 : 1 6 , dlugosc: 8  
 7 : 1 7 , dlugosc: 2  
 8 : 1 6 3 8 , dlugosc: 16

Rozwiązania z części A:

**A1**

kraw.	działanie	1	2	3	4	aktualnie dodane
–	(r,p)	1,-	2,-	3,-	4,-	
(1,2)	dodajemy	1,-	1,1	3,-	4,-	(1,2)
(1,4)	dodajemy	1,-	1,1	3,-	1,1	(1,2)(1,4)
(2,3)	dodajemy	1,-	1,1	1,2	1,1	(1,2)(1,4)(2,3)
Teraz zapisujemy drzewo						
(2,3)	usuwamy	1,-	1,1	3,-	1,1	(1,2)(1,4)
(3,4)	dodajemy	1,-	1,1	1,4	1,1	(1,2)(1,4)(3,4)
Teraz zapisujemy drzewo						
(3,4) była ostatnia na liście, więc usuwamy dwie						
(3,4)	usuwamy	1,-	1,1	3,-	1,1	(1,2)(1,4)
(1,4)	usuwamy	1,-	1,1	3,-	4,-	(1,2)
Po usunięciu bierzemy kolejną po usuniętej na liście						
(2,3)	dodajemy	1,-	1,1	1,2	4,-	(1,2)(2,3)
(3,4)	dodajemy	1,-	1,1	1,2	1,3	(1,2)(2,3)(3,4)
Teraz zapisujemy drzewo						
(3,4) była ostatnia na liście, więc usuwamy dwie						
(3,4)	usuwamy	1,-	1,1	1,2	4,-	(1,2)(2,3)
(2,3)	usuwamy	1,-	1,1	3,-	4,-	(1,2)
Po usunięciu bierzemy kolejną po usuniętej na liście						
(3,4)	dodajemy	1,-	1,1	3,-	3,3	(1,2)(3,4)
(3,4) była ostatnia na liście, więc usuwamy dwie						
(3,4)	usuwamy	1,-	1,1	3,-	4,-	(1,2)
(1,2)	usuwamy	1,-	2,-	3,-	4,-	
Po usunięciu bierzemy kolejną po usuniętej na liście						
(1,4)	dodajemy	1,-	2,-	3,-	1,1	(1,4)
(2,3)	dodajemy	1,-	2,-	2,2	1,1	(1,4)(2,3)
(3,4)	dodajemy	1,-	<b>1,3!!</b>	1,4	1,1	(1,4)(2,3)(3,4)
Teraz zapisujemy drzewo						
(3,4) była ostatnia na liście, więc usuwamy dwie						
(3,4)	usuwamy	1,-	3,3	3,-	1,1	(1,4)(2,3)
(2,3)	usuwamy	1,-	2,-	3,-	1,1	(1,4)
Po usunięciu bierzemy kolejną po usuniętej na liście						
(3,4)	dodajemy	1,-	2,-	1,4	1,1	(1,4)(3,4)
(3,4) była ostatnia na liście, więc usuwamy dwie						
(3,4)	usuwamy	1,-	2,-	3,-	1,1	(1,4)
(1,4)	usuwamy	1,-	2,-	3,-	4,-	
Po usunięciu bierzemy kolejną po usuniętej na liście						
(2,3)	KONIEC, Tree(1)=END					

Uzyskane drzewa (w kolejności)



**A2**

kraw.	działanie	1	2	3	4	5	6	7	8	9	10	11	12	13	14
–	(r,p)	1,-	1,3	1,1	1,3	5,-	6,-	1,8	1,1	1,8	6,6	5,5	5,11	5,5	5,13
(3,12)	dodana	1,-	1,3	1,1	1,3	1,11	6,-	1,8	1,1	1,8	6,6	1,12	1,3	1,5	1,13
(7,14)	nic	1,-	1,3	1,1	1,3	1,11	6,-	1,8	1,1	1,8	6,6	1,12	1,3	1,5	1,13
(6,7)	dodana	1,-	1,3	1,1	1,3	1,11	1,7	1,8	1,1	1,8	1,6	1,12	1,3	1,5	1,13
Teraz zapisujemy drzewo															
(6,7)	usuwana	1,-	1,3	1,1	1,3	1,11	6,-	1,8	1,1	1,8	6,6	1,12	1,3	1,5	1,13
(7,10)	dodana	1,-	1,3	1,1	1,3	1,11	1,10	1,8	1,1	1,8	1,7	1,12	1,3	1,5	1,13
Teraz zapisujemy drzewo															
(7,10)	usuwana	1,-	1,3	1,1	1,3	1,11	10,10	1,8	1,1	1,8	10,-	1,12	1,3	1,5	1,13
Teraz byśmy sięgnęli po kolejną (po $v_7v_{10}$ ) kraw. z listy ...															

