

Algorytmy grafowe 03: Spójność.

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

Zadanie A.1. Korzystając z algorytmu scalania wyznacz wszystkie składowe spójności grafu zadanego macierzą przyległości. W rozwiązaniu:

- Zapisz wszystkie kolejne macierze powstałe w trakcie działania algorytmu.
- W macierzach zaznacz kolumny (wiersze), które zostaną scalone w kolejnym kroku
- oraz zapisz, które wierzchołki są przyporządkowane do poszczególnych kolumn (wierszy).
- Na końcu wypisz składowe spójności.

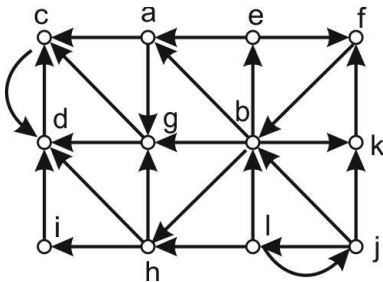
Wierzchołki są oznaczone kolejno (zgodnie z kolejnością wierszy) alfabetycznie: a, b, c, ...

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Zadanie A.2. Korzystając z algorytmu Leifmanna wyznacz wszystkie składowe silnej spójności w grafie na poniższym rysunku. Dla każdej iteracji (po każdej zmianie przez algorytm) zapisz:

- Uaktualnione L i C ;
- Aktualnie rozpatrywany \bar{V} ;
- Aktualnie rozpatrywany wierzchołek z \bar{V} i dla niego V_{00} , V_{01} , V_{10} i V_{11} (nawet jeśli są puste)

Przy wyborze wierzchołka z \bar{V} kieruj się kolejnością alfabetyczną.



B Program do napisania

Proszę o przesłanie

- do nocy z soboty na niedzielę (**4/5 kwietnia**);
- w mailu o tytule **AGR02** (WAŻNE: Nie będę czytała tych maili, więc z istotnymi sprawami proszę się zgłaszać w osobnych mailach.)
- plików o zindywidualizowanej nazwie **02NazwiskoImie.py** albo **02NazwiskoImie.txt** (jeśli .py nie chce się wysłać) albo skompresowane o nazwie **02NazwiskoImie** (ale TYLKO jeśli piszą Państwo w kilku plikach) albo **02NazwiskoImieNieDziała.*** (jeśli podjęł 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.**
- Proszę o wpisanie w programie '*graph03.txt*' a nie odwołania do pliku, które Państwo wykorzystywali.
- Proszę nie wysyłać mi pliku tekstowego z grafem.

UWAGA: Przypominam, że piszemy w Pythonie**3**

Definicja. Krawędź e w grafie G jest krawędzią cięcia, jeśli $G - e$ ma więcej składowych spójności niż G .

Zadanie B.1. W pliku *graph03.txt* zapisana jest macierz przyległości grafu. Dla Państwa wygody w tej macierzy znaki „0” zostały zastąpione znakiem „-”, żeby wygodnie skorzystać bez zbytnich przeróbek z zadania z poprzednich zajęć. We wczytanym grafie wierzchołki powinny być numerowane kolejno liczbami naturalnymi zaczynając od 1 (zgodnie z kolejnością wierszy).

Napisz program, który wykorzystując algorytm BFS dla każdej krawędzi określa, czy jest krawędzią cięcia.

W wyjściu powinny się znajdować w kolejnych liniach: krawędź i TAK (jeśli krawędź jest krawędzią cięcia) albo NIE (jeśli krawędź nie jest krawędzią cięcia.)

WSKAZÓWKI: Można „wygenerować” listę krawędzi (jako listę) i listę następników (jako słownik). Sprawdzić, czy krawędź jest krawędzią cięcia można na przykład tak:

- Usuwamy krawędź vw z grafu.
- Zaczynamy BFS z v . Jeśli w trakcie odkrywamy w , to vw nie jest krawędzią cięcia. W p.p. jest krawędzią cięcia. Warto zauważyć, że nie musimy zapisywać ani listy odkrytych wierzchołków ani krawędzi.

PYTANIE: A dlaczego to da poprawną odpowiedź?

Uwaga: warto pobrać plik *txt* ze strony a nie kopiować to poniżej do pliku *txt*.

WEJŚCIE:

```

---- 1 -- 1 -----
----- 1 -- 1 -----
--- 1 1 ----- 1 --
-- 1 -----
1 - 1 ---- 1 -----
----- 1 1 - 1 --
- 1 ----- 1 -----
1 --- 1 -----
----- 1 -- 1 --
- 1 ---- 1 ----- 1 ---
----- 1 -- 1 -- 1 -
----- 1 ---- 1 -----
----- 1 -----
-- 1 -- 1 -- 1 -----
----- 1 ---- 1
----- 1 -

```

WYJŚCIE:

```

(1, 5) NIE
(1, 8) NIE
(2, 7) NIE
(2, 10) NIE
(3, 4) TAK
(3, 5) TAK
(3, 14) TAK
(5, 8) NIE
(6, 11) NIE
(6, 12) NIE
(6, 14) NIE
(7, 10) NIE
(9, 11) NIE
(9, 14) NIE

```

(10, 13) TAK
(11, 12) NIE
(11, 15) TAK
(15, 16) TAK

Przykładowe rozwiązanie **A.1**

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>
<i>a</i>	0	0	1	0	0	0	0	0	0	1
<i>b</i>	0	0	0	1	0	1	0	1	0	0
<i>c</i>	1	0	0	0	0	0	0	0	0	1
<i>d</i>	0	1	0	0	0	1	0	0	0	0
<i>e</i>	0	0	0	0	0	0	1	0	1	0
<i>f</i>	0	1	0	1	0	0	0	0	0	0
<i>g</i>	0	0	0	0	1	0	0	0	1	0
<i>h</i>	0	1	0	0	0	0	0	0	0	0
<i>i</i>	0	0	0	0	1	0	1	0	0	1
<i>j</i>	1	0	1	0	0	0	0	0	1	0

scalamy *a*, *c*, *j*

	<i>acj</i>	<i>b</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>
<i>acj</i>	1	0	0	0	0	0	0	1
<i>b</i>	0	0	1	0	1	0	1	0
<i>d</i>	0	1	0	0	1	0	0	0
<i>e</i>	0	0	0	0	0	1	0	1
<i>f</i>	0	1	1	0	0	0	0	0
<i>g</i>	0	0	0	1	0	0	0	1
<i>h</i>	0	1	0	0	0	0	0	0
<i>i</i>	1	0	0	1	0	1	0	0

scalamy *acj*, *i*

	<i>acji</i>	<i>b</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
<i>acji</i>	1	0	0	1	0	1	0
<i>b</i>	0	0	1	0	1	0	1
<i>d</i>	0	1	0	0	1	0	0
<i>e</i>	1	0	0	0	0	1	0
<i>f</i>	0	1	1	0	0	0	0
<i>g</i>	1	0	0	1	0	0	0
<i>h</i>	0	1	0	0	0	0	0

scalamy *acji*, *e*, *g*

	<i>acjieg</i>	<i>b</i>	<i>d</i>	<i>f</i>	<i>h</i>
<i>acjieg</i>	1	0	0	0	0
<i>b</i>	0	0	1	1	1
<i>d</i>	0	1	0	1	0
<i>f</i>	0	1	1	0	0
<i>h</i>	0	1	0	0	0

$J = \{1\}$, więc usuwamy pierwszy wiersz i kolumnę.

Zapisujemy składową spójności: $\{a, c, j, i, e, g\}$

	<i>b</i>	<i>d</i>	<i>f</i>	<i>h</i>
<i>b</i>	0	1	1	1
<i>d</i>	1	0	1	0
<i>f</i>	1	1	0	0
<i>h</i>	1	0	0	0

Scalamy *b*, *d*, *f*, *h*

	<i>bdfh</i>
<i>bdfh</i>	1

$J = \{1\}$, więc usuwamy pierwszy wiersz i kolumnę.

Zapisujemy składową spójności: $\{b, d, f, h\}$

Przykładowe rozwiązanie **A.2**

$L : \{a, b, c, d, e, f, g, h, i, j, k, l\}, C : \emptyset$

$\bar{V} : \{a, b, c, d, e, f, g, h, i, j, k, l\}$

rozpatrujemy a

wierzchołki osiągalne z a : $\{c, d, g\}$

wierzchołki, z których a jest osiągalny: $\{b, e, f, j, k, l\}$

$V_{11} = \emptyset, V_{10} = \{g, c, d\}, V_{01} = \{b, e, f, j, k, l\}, V_{00} = \{a, h, i\}$

$L : \{h, i\}, \{c, d, g\}, \{b, e, f, j, k, l\}; C : \{a\};$

$\bar{V} = \{h, i\},$

h – ma stopień wejścia 0

$L : \{i\}, \{c, d, g\}, \{b, e, f, j, k, l\}; C : \{a\}, \{h\};$

$\bar{V} = \{i\};$

i – ma stopień wejścia i wyjścia 0;

$L : \{c, d, g\}, \{b, e, f, j, k, l\}; C : \{a\}\{h\}\{i\}$

$\bar{V} = \{c, d, g\}$

g – ma stopień wejścia 0;

$L : \{c, d\}, \{b, e, f, j, k, l\}; C : \{a\}\{h\}\{i\}\{g\}$

$\bar{V} = \{c, d\}$

rozpatrujemy c :

$V_{11} = \{c, d\}, V_{10} = V_{01} = V_{00} = \emptyset$

$L : \{b, e, f, j, k, l\}; C : \{a\}\{h\}\{i\}\{g\}\{c, d\}$

$\bar{V} = \{b, e, f, j, k, l\}$

rozpatrujemy b

$V_{11} = \{b, e, f, k\}, V_{10} = \emptyset, V_{01} = \{j, l\}, V_{00} = \emptyset$

$L : \{j, l\}; C : \{a\}\{h\}\{i\}\{g\}\{c, d\}\{b, e, f, k\}$

$\bar{V} = \{j, l\}$

$V_{11} = \{j, l\}, V_{10} = V_{01} = V_{00} = \emptyset$

$L : \emptyset; C : \{a\}\{h\}\{i\}\{g\}\{c, d\}\{b, e, f, k\}\{j, l\}$