Vzor semestrální práce z předmětu Y1TG – Teorie grafů

David Lister

April 15, 2021

1 Úvod

Toto je vzor semestrální práce z předmětu Teorie grafů*. Není nutné se ho držet do posledního puntíku, ale vaše semestrální práce by měla obsahovat všechny zde uvedené body.*

1.1 Zadání

Cílem semestrální práce je vytvořit funkci, která vygeneruje matici sousednosti neorientovaného neohodnoceného grafu.

1.2 Popis algoritmu

Použil jsem standardní algoritmus, který jsme probírali na přednáškách.

nebo

Algoritmus jsem převzal z publikace [1]. Algoritmus je založen na [...]. *Následuje popis algoritmu.*

případně ještě

V algoritmu jsem provedl tyto úpravy: [...]

2 Implementace

Algoritmus jsem implementoval v prostředí Matlab R2020b. *(nebo Python, C++, ... – co se vám zlíbí.)*

2.1 Popis vstupu

Na vstupu je neorientovaný neohodnocený prostý graf popsaný seznamem hran.

Vrcholy grafu jsou označeny přirozenými čísly počínaje jedničkou.

Seznam hran je neprázdná matice `hrany` o dvou sloupcích. Každý řádek matice obsahuje dvě přirozená čísla, označující koncové vrcholy jedné hrany.

Počet vrcholů odpovídá největšímu číslu, které se vyskytuje v seznamu hran. Pokud některé číslo mezi jedničkou a maximem v seznamu není, je považováno za izolovaný vrchol.

2.2 Popis výstupu

Výstupem je matice sousednosti daného grafu.

Pokud je výstupem něco, co není v teorii grafů jednoznačně definováno, musí být výstup popsán podrobněji, například:

Výstupem je trojice `(x,A,chi)`, kde

- `x` je reálné číslo z intervalu $\langle 0, 1\rangle$, udávající pravděpodobnost, že [...],

- `A` je matice $m \times 3$, kde m je počet vrcholů grafu. V $i-$tém řádku matice je popořadě uložen vstupní, výstupní a celkový stupeň $i-$tého vrcholu,
- `chi` je chromatické číslo grafu.

2.3 Výpis funkce

```
function A = adjMatrix( hrany )
%
% sestroji matici sousednosti grafu
% VSTUP: matice (m,2) obsahujici seznam hran
% VYSTUP: matice sousednosti
% (c) Dave Lister

n = max ( max ( hrany ) ) ;     % nejvetsi cislo vrcholu = pocet vrcholu
A = zeros ( n );                % matice sousednosti, zatim bez hran

m = size ( hrany, 1 );          % pocet hran

for i = 1 : m                   % prochazim vsechny hrany
    A ( hrany(i,1) , hrany(i,2) ) = 1;
    A ( hrany(i,2) , hrany(i,1) ) = 1;
    % neorientovany graf - oba smery
end
end
```

2.4 Způsob volání funkce

Funkce se volá příkazem » `I = adjMatrix (H)`, kde `H` je matice podle specifikace uvedené v odstavci 2.1.

2.5 Omezení

Funkce nekontroluje správnost zadání, speciálně nekontroluje, jestli

- matice `hrany` není prázdná
- matice `hrany` má dva sloupce
- matice `hrany` obsahuje pouze přirozená čísla
- graf je prostý.

3 Testování

Pro testování správnosti funkce jsem použil následující sadu testovacích příkladů:

3.1 Testovací příklad 1

První graf z obrázku 1.

```
>> H = [ 1 3 ; 3 4 ; 4 1 ; 2 3 ]

H =

     1     3
     3     4
     4     1
```

```
     2     3

>> A = adjMatrix (H)

A =

     0     0     1     1
     0     0     1     0
     1     1     0     1
     1     0     1     0
```

Test proběhl bez chyb.

3.2 Testovací příklad 2

Druhý graf z obrázku 1 – zkouším, jestli zvládne smyčku.

```
>> H = [2 3 ; 1 1 ; 3 1]

H =

     2     3
     1     1
     3     1

>> A = adjMatrix (H)

A =

     1     0     1
     0     0     1
     1     1     0
```

Test proběhl bez chyb.

3.3 Testovací příklad 3

Třetí graf z obrázku 1 – izolovaný vrchol.

```
>> H = [ 5 2 ; 1 5 ; 1 4 ; 5 4 ]

H =

     5     2
```

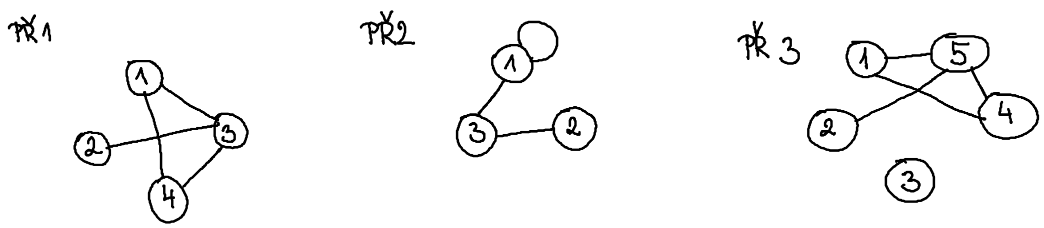

Figure 1: Testovací příklady 1 – 3.

```
     1     5
     1     4
     5     4

>> A = adjMatrix (H)

A =

     0     0     0     1     1
     0     0     0     0     1
     0     0     0     0     0
     1     0     0     0     1
     1     1     0     1     0
```

Test proběhl bez chyb.

3.4 Další testovací příklady

Celkem by mělo být 3 až 5 testovacích příkladů menší velikosti (do 10 vrcholů).

Dobře zvolené příklady testují, jestli funkce opravdu dělá, co má, a co slibuje její popis. Nejsou úplně triviální a zahrnují všechny neobvyklé, vyjímečné a krajní případy, které vás napadnou. Například nesouvislý graf, neexistence hledaného objektu (cesta z vrcholu v do vrcholu u), existence více hledaných objektů, divný požadavek (najdi cestu z vrcholu v do vrcholu v), ...

Věci, které funkce neumí (chybný formát vstupu, graf bez nějaké vlastnosti a pod.) musejí být popsány v odstavci 2.5.

4 Příklad použití

Zde by měl být nějaký hezký příklad, na kterém lze ukázat, jak je váš prográmek šikovný a co všechno umí.

Rozhodně by měl být trochu větší, než testovací příklady, tak 15 – 25 vrcholů; záleží na tom, jaký typ úlohy řešíte. Graf by měl být co nejobecnější, jaký vaše funkce zvládne – pokud například umíte najít nějakou cestu v ohodnoceném grafu, který obsahuje cykly záporné délky, pak nemá smysl demonstrovat vaše umění na acyklickém grafu s nezáporně ohodnocenými hranami.

5 Závěr

Tuhle semestrálku mi byl čert dlužný.

References

[1] Šu-ej Ču-Žou, Kung Pao: *On the Adjacency Matrix.* Some Journal **33** (1992), 503–535.