MMAD - Úkol 2

Filip Ditrich

Unicorn University, Prague, Czech Republic
27. dubna 2024

Úloha 1 1
Úloha 2 5
Úloha 3 6
Úloha 4 7
Úloha 5 8
Úloha 6 9
Úloha 7 10
Úloha 8 11
Úloha 9 12
Úloha 10. 13
Úloha 11. 14
Úloha 12. 15
Úloha 13. 16
Úloha 14. 17
Úloha programovací 1 18
Úloha programovací 2 19

Úloha 1 (2 body)

Pro následující grafy určete minimální stupeň $\delta(G)$, maximální stupeň $\Delta(G)$, skóre grafu a barevnost pro následující grafy:

1. a) Cesta P_n
2. b) Kružnice C_n
3. c) Úplný graf K_n
4. d) Úplný bipartitní graf $K_{m,n}$
5. e) Papův graf ukázaný níže

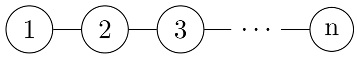
Řešení

Nejprve si připomeneme definice:

- **Minimální stupeň** $\delta(G)$ je nejmenší stupeň vrcholu v grafu G.
- **Maximální stupeň** $\Delta(G)$ je největší stupeň vrcholu v grafu G.

- **Skóre grafu** je součet stupňů všech vrcholů v grafu G.
- **Barevnost** je minimální počet barev potřebných k obarvení vrcholů grafu G tak, aby žádné dva sousední vrcholy neměly stejnou barvu.

1a) Cesta P_n Definice: Cesta P_n je graf, který má n vrcholů a $n-1$ hran.



Obrázek (1) – Cesta P_n

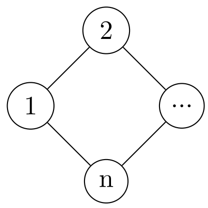
ODPOVĚĎ 1A

Uvažujme cestu P_n s $n \geq 3$ vrcholy.

- $\delta(P_n) = 1$ ... první a poslední vrchol mají stupeň 1
- $\Delta(P_n) = 2$ ... všechny vrcholy kromě prvního a posledního mají stupeň 2
- Skóre grafu $P_n = 2n - 2$ ... vnitřní vrcholy mají stupeň 2, první a poslední vrchol mají stupeň 1
- Barevnost grafu $P_n = 2$ ... vždy stačí 2 barvy

Pro P_2 pak platí, že $\delta(P_2) = \Delta(P_2) = 1$.

1b) Kružnice C_n Definice: Kružnice C_n má n vrcholů a každý vrchol je spojen s předchozím a následujícím vrcholem.



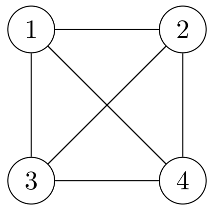
Obrázek (2) – Kružnice C_n

ODPOVĚĎ 1B

Uvažujme kružnici C_n s $n \geq 3$ vrcholy.

- $\delta(C_n) = 2$ ... všechny vrcholy mají stupeň 2
- $\Delta(C_n) = 2$ ... všechny vrcholy mají stupeň 2
- Skóre grafu $C_n = 2n$ ... všechny vrcholy mají stupeň 2
- Barevnost grafu $C_n = 2$ pro sudé n, 3 pro liché n

1c) Úplný graf K_n Definice: Úplný graf K_n má n vrcholů a každý vrchol je spojen s každým jiným vrcholem. Speciální případ úplného grafu je úplný graf K_2, který má 2 vrcholy a 1 hranu a jedná se tedy o cestu P_2. Další speciální případ je úplný graf K_3, který má 3 vrcholy a 3 hrany a jedná se tedy o kružnici C_3.



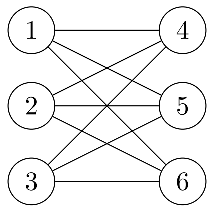
Obrázek (3) − Úplný graf K_4

ODPOVĚĎ 1C

Uvažujme úplný graf K_n s $n \geq 3$ vrcholy.

- $\delta(K_n) = n - 1$ ... všechny vrcholy mají stupeň $n - 1$
- $\Delta(K_n) = n - 1$ ... všechny vrcholy mají stupeň $n - 1$
- Skóre grafu $K_n = n(n - 1)$ ... všechny vrcholy mají stupeň $n - 1$
- Barevnost grafu $K_n = \begin{cases} n - 1 & \text{pro sudé } n \\ n & \text{pro liché } n \end{cases}$

1d) Úplný bipartitní graf $K_{m,n}$ Definice: Úplný bipartitní graf $K_{m,n}$ má m vrcholů v jedné partitě a n vrcholů v druhé partitě a každý vrchol z jedné partity je spojen s každým vrcholem z druhé partity.



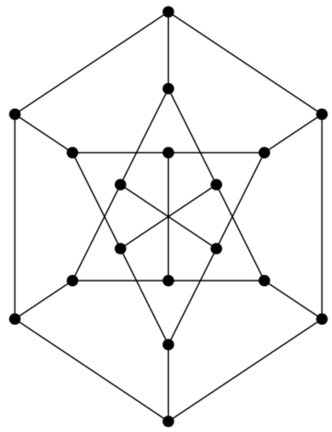
Obrázek (4) − Úplný bipartitní graf $K_{3,3}$

ODPOVĚĎ 1D

Uvažujme úplný bipartitní graf $K_{m,n}$ s $m,n \geq 1$ vrcholy.

- $\delta(K_{m,n}) = n$ ... všechny vrcholy z první partity mají stupeň n
- $\Delta(K_{m,n}) = m$ ... všechny vrcholy z druhé partity mají stupeň m
- Skóre grafu $K_{m,n} = mn$ ... všechny vrcholy z první partity mají stupeň n, všechny vrcholy z druhé partity mají stupeň m
- Barevnost grafu $K_{m,n} = 2$ ... vždy stačí 2 barvy, jedna pro každou partitu

1e) Papův graf Definice: Na obrázku níže je zobrazen Papův graf s 18 vrcholy a 27 hranami.



Obrázek (5) − Papův graf

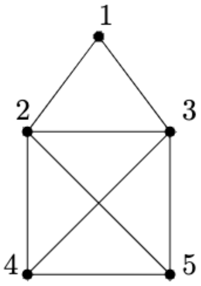
ODPOVĚĎ 1E

TODO:

- δ(Papův graf) = 3 ... všechny vrcholy mají stupeň 3
- Δ(Papův graf) = 3 ... všechny vrcholy mají stupeň 3
- Skóre grafu Papův graf = $18 \times 3 = 54$ ... všechny vrcholy mají stupeň 3
- Barevnost grafu Papův graf = $TODO$?

Úloha 2 (2 body)

Máme graf $G = (V, E)$. Pro ukázku si představme následující graf H:



Obrázek (6) – Graf H

Definujme Laplacovu matici L jako:

$$L_{i,j} = \begin{cases} \deg(v_i) & \text{pro } i = j \\ -1 & \text{pro } i \neq j \text{ a } v_i \text{ je spojen s } v_j \\ 0 & \text{jinak} \end{cases}$$

Například pro ukázkový graf dostaneme Laplacovu matici:

$$L = \begin{pmatrix} 2 & -1 & -1 & 0 & 0 \\ 1 & 4 & -1 & -1 & -1 \\ -1 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{pmatrix}$$

Mějme k-regulární graf $G = (V, E)$ velikosti $|V| = n$, tj. graf pro který víme, že $\deg(v_i) = k$ pro všechny vrcholy $v_i \in V$. Navíc víme, že vlastní čísla matice sousednosti jsou reálná v pořadí $\lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_n(A)$. Lze nějak obecně vyjádřit všechna vlastní čísla Laplacovy matice takového grafu?

Řešení

TODO

Úloha 3 (2 body)

Určete minimální a maximální počet hran v grafu na n vrcholech s c komponentami.

Řešení

Cílem je nalézt rozsah (minimální a maximální) možných hran v grafu skládajícím se z n vrcholů rozdělených do c komponent. Komponentou se rozumí podgraf, ve kterém jsou všechny vrcholy spojeny cestou a který není spojen s žádnými dalšími vrcholy v hlavním grafu.

Postup pro nalezení minimálního počtu hran:

- Izolované komponenty: Minimální počet hran nastává, když mají komponenty co nejméně hran. Extrémním případem je mít komponenty bez hran, tedy izolované vrcholy.
- Neizolované komponenty: Pro každou komponentu, která není jediný izolovaný vrchol, je minimální struktura vlastně strom. Strom s k vrcholy má $k-1$ hran (minimum pro udržení grafu spojeného).
- Pokud je potřeba c komponent a předpokládáme že $c-1$ komponent jsou jednotlivé izolované vrcholy a jedna komponenta obsahuje zbytek vrcholů, $n-(c-1)$, tato poslední komponenta jako strom by měla $n-(c-1)-1 = n-c$ hran.
- Minimální počet hran je tedy $0+(n-c) = n-c$.

Příklad na grafu s $n = 10$ vrcholy a $c = 3$ komponentami:

- Izolované komponenty: 2 izolované vrcholy, 1 komponenta s 8 vrcholy.
- Minimální počet hran: $10-3 = 7$.

ODPOVĚĎ 3A

Minimální počet v grafu na n vrcholech s c komponentami je $n-c$.

Postup pro nalezení maximálního počtu hran:

- Maximální počet hran nastává, když každá komponenta je úplný graf (graf, kde jsou každé různé vrcholy spojeny jedinou hranou).
- Respektive postačí nám jedna komponenta jako úplný graf a zbytek komponent jako izolované vrcholy (tedy bez hran).
- Počet hran v úplném grafu na k vrcholech je $\binom{k}{2}$
- Náš úplný graf má $n-(c-1)$ vrcholů, tedy počet vrcholů mínus počet ostatních izolovaných vrcholů (komponent).
- Maximální počet hran je tedy $\binom{n-(c-1)}{2}$.

Příklad na grafu s $n = 10$ vrcholy a $c = 3$ komponentami:

- Úplný graf s 8 $(10-(3-1))$ vrcholy a 1 izolovaný vrchol.
- Maximální počet hran: $\binom{10-2}{2} = 28$.

ODPOVĚĎ 3B

Maximální počet v grafu na n vrcholech s c komponentami je $\binom{n-(c-1)}{2}$.

Úloha 4 (2 body)

Mějme následující funkci:

$$f(x,y) = 100(y - x^2)^2 + (1 - x)^2$$

Vypočtěte gradient $\nabla f(x)$ a Hessian $\nabla^2 f(x)$ a rozhodněte (a zdůvodněte) zda-li je v bodě $(1,1)$ splněna 1. podmínka pro lokální minimizátor (nulovost gradientu) či zda-li je splněna i 2. podmínka pro Hessovu matici.

Řešení

TODO

Úloha 5

(2 body)

Mějme množinu $\{1, 2, \dots, n\}$. Určete, kolik je možné na této množině najít různých kružnic délky n? (jedná se tedy o počet průchodů, ale neorientovaného grafu).

Řešení

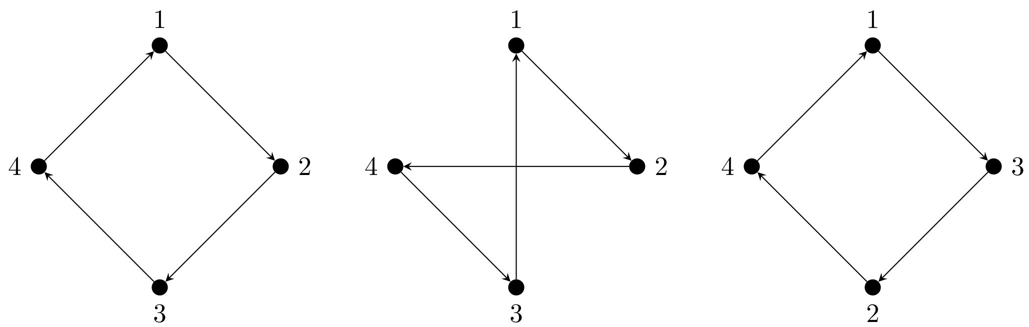
Problém můžeme řešit následovně:

- Krok 1: Seřadíme vrcholy kružnice do pořadí $1, 2, \dots, n$. Takových sekvencí je $n!$.
- Krok 2: Zvolíme si jeden výchozí vrchol (symetrická rotace). Tím tedy získáme $(n-1)!$ unikátních sekvencí, ignorujeme-li rotace.
- Krok 3: Otočením sekvence získáme stejnou kružnici. Počet kružnic tedy musíme dělit dvěma (pro $n > 2$), protože každá sekvence a její zrcadlový obraz jsou v kružnici identické.
- Počet různých neorientovaných kružnic délky n (pro $n > 2$) je tedy: $\frac{(n-1)!}{2}$.

Ukázka: na příkladu množiny $\{1, 2, 3, 4\}$:
Dle definice výše víme, že počet různých neorientovaných kružnic délky 4 bude $\frac{(4-1)!}{2} = 3$.

A bude se jednat o tyto 3 kružnice:

1. $1 \to 2 \to 3 \to 4 \to 1$
2. $1 \to 2 \to 4 \to 3 \to 1$
3. $1 \to 3 \to 2 \to 4 \to 1$



Obrázek (7) — Různé neorientované kružnice délky 4

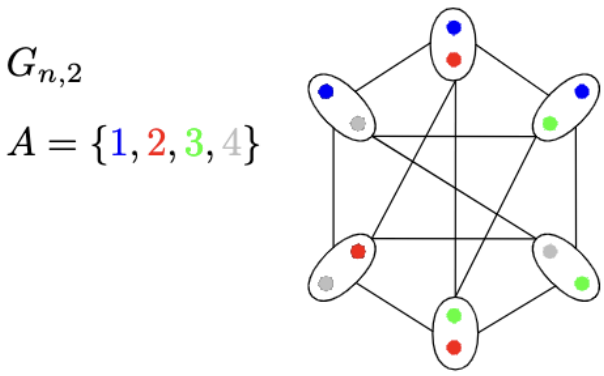
ODPOVĚĎ

Pro množinu $\{1, 2, \dots, n\}$ je možné najít $\frac{(n-1)!}{2}$ různých neorientovaných kružnic délky n.

Úloha 6

(2 body)

Mějme graf $G_{n,2} = (V, E)$ definovaný následovně. Množina vrcholů jsou všechny podmnožiny množiny $A = \{1, 2, \dots, n\}$ o velikosti 2, tedy například $v_1 = \{1, 2\}, v_2 = \{2, 3\}, \dots$. Hrany spojují ty vrcholy $v_i = \{a, b\}, v_2 = \{c, d\}$, které sdílí právě jeden prvek, tj. $a = b \neq c = d$. Příklad takového grafu je vidět na následujícím obrázku.



Obrázek (8) — Graf $G_{n,2}$

Pro takový obecný graf $G_{n,2}$ určete jaký bude jeho minimální a maximální stupeň vrcholu vyjádřeno jako funkce n . Také určete počet hran tohoto grafu, opět jako funkci n.

Řešení

TODO

Úloha 7 (2 body)

Zdůvodněte, proč každá hrana vrcholově 2-souvislého grafu musí ležet na kružnici.

Řešení

TODO

Úloha 8 (2 body)

Určete vrcholový a hranový stupeň grafu, neboli $\alpha(G)$ a $\kappa(G)$, pro následující grafy:

1. Cesta P_n
2. Kružnice C_n
3. Úplný graf K_n
4. Úplný bipartitní graf $K_{m,n}$

Řešení

TODO

Úloha 9 (2 body)

Vezměmě si grafy typu strom o fixní velikosti n. Rozhodněte a nakreslete, jaký strom o velikosti n má:

1. Největší hodnotu nezávislosti $\alpha(G)$
2. Nejmenší hodnotu nezávislosti $\alpha(G)$
3. Největší hodnotu vrcholového pokrytí $\beta(G)$
4. Nejmenší hodnotu vrcholového pokrytí $\beta(G)$

Řešení

TODO

Úloha 10

(2 body)

Ukažte proč pro každý kubický graf G, t.j. takový, že všechny stupně vrcholů jsou 3, platí, že stupeň vrcholové i hranové souvislosti se rovnají, tj. $\alpha(G) = \kappa(G)$. *Hint: Pokuste se rozebrat případy pro různé vrcholové stupně souvislosti.*

Řešení

TODO

Úloha 11 (2 body)

Pokuste se navrhnout Turingův stroj pro rozpoznání, že neorientovaný graf má izolovaný vrchol.
Hint: Graf uložte na pásku jako matici sousednosti (nezapomeňte na oddělovače řádků) a v ní pomocí pravidel nalezněte takový vrchol.

Řešení

TODO

Úloha 12 (2 body)

Využijte vysvětlení proč platí polynomialita 2-SAT a nakreslete graf odpovídající následující formuli a otestujte a případně ukažte, zda-li je splněna. *Pozn.: Graf na kreslete, i když budete schopni splnitelnost rozhodnout jinak.*

$$f(x_1, x_2, x_3, x_4, x_5) = (x_1 \vee \overline{x_3}) \wedge (x_2 \vee x_3) \wedge (\overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee \overline{x_4}) \wedge (x_4 \vee x_5) \wedge (\overline{x_3} \vee x_5)$$

Řešení

TODO

Úloha 13 (2 body)

Na základě vysvětlení převodu SAT na IND nakreslete graf odpovídající následující formuli a otestujte splnitelnost formule nalezením nezávislé množiny. *Pozn.: Graf nakreslete a zhodnoťte, i když budete schopni splnitelnost rozhodnout jinak.*

$$f(x_1, x_2, x_3, x_4, x_5) = (x_2 \vee \overline{x_3} \vee x_4) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_3} \vee x_4) \wedge (x_1 \vee \overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (x_4 \vee x_5)$$

Řešení

TODO

Úloha 14 (2 body)

Na základě vysvětlení převodu 3-SAT na 3-COL nakreslete graf odpovídající následující formuli a otestujte barevnost grafu.

$$f(x_1, x_2, x_3, x_4, x_5) = (x_1 \vee x_2 \vee \overline{x_4}) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_2} \vee \overline{x_3} \vee x_4) \wedge (x_1 \vee \overline{x_2} \vee x_3)$$

Řešení

TODO

Úloha programovací 1 (4 body)

Naprogramujte algoritmus pro testování isomorfismu grafů hrubou silou a otestujte to na pár příkladech grafů s využitím knihovní funkce pro testování isomorfismu.

Řešení

Isomorfismus lze testovat hrubou silou, kde se všechny možné permutace uzlů grafu G porovnají s uzly grafu H.

Vytvoříme funkci `isomorphism(G, H)`, která otestuje isomorfismus grafů G a H následovně:

- Pokud mají grafy různý počet uzlů, vrátí `False`.
- Pro všechny permutace uzlů grafu G:
  - Vytvoří mapování uzlů grafu G na uzly grafu H.
  - Pokud všechny uzly grafu G mají svůj ekvivalent v grafu H a všechny hrany zůstanou zachovány, vrátí `True`.
- Pokud žádná permutace nevyhovuje, vrátí `False`.

Implementačně je algoritmus následující:

```python
import networkx as nx
import itertools

def isomorphism(G, H):
    if len(G.nodes) != len(H.nodes):
        return False

    # získání všech permutací uzlů grafu G
    perms = itertools.permutations(list(G.nodes))
    for perm in perms:
        # vytvoření mapování uzlů grafu G na uzly grafu H
        # pokud všechny uzly grafu G mají svůj ekvivalent v grafu H a všechny hrany zůstanou zachovány, vrátí True
        mapping = dict(zip(G.nodes, perm))
        has_all_nodes = all([mapping[u] in H.nodes for u in G.nodes])
        has_all_edges = all([mapping[u] in H.nodes for u in G.nodes])
        if has_all_nodes and has_all_edges:
            return True
        return False
```

Poté již zbývá jen funkci výše otestovat na několika příkladech grafů a porovnat výsledky s knihovní funkcí pro testování isomorfismu:

```python
G = nx.generators.small.cycle_graph(5)
H = nx.complement(nx.generators.small.cycle_graph(5))

my_isom = isomorphism(G, H)
nx_isom = nx.is_isomorphic(G, nx.complement(H))
print(f"Vlastní: {my_isom}\nNetworkX: {nx_isom}\nÚspěch: {'Ano' if my_isom == nx_isom else 'Ne'}")
```

Úplný zdrojový kód je k nalezení v souboru ukol-2-k1.py.

Úloha programovací 2

(4 body)

Realizujte hrubou silou nalezení největší nezávislé množiny daného grafu a následně otestujte, že je množina nezávislá. Následně se pokuste vylepšit řešení procházení množinami použitím sousedů vrcholu.

Pokud chceme testovat procházení, nabízí se, ne nutně, řešení pomocí nějakého rekurzivního přístupu.

Řešení

TODO