Otázky k druhému úkolu #2

Matematika a algoritmy pro datamining
Unicorn University

1. března 2023

Všechny své odpovědi čitelně a pečlivě popište a zdůvodněte (odpověď bez zdůvodnění se za odpověď nepočítá). Pokud to není výslovně uvedeno, není povolené k řešení používat počítač. Čísla vystupující z počítače vždy uvádějte, tak aby bylo jasné, proč jste dospěli k daným závěrům. V následující části jsou dva typy zadání

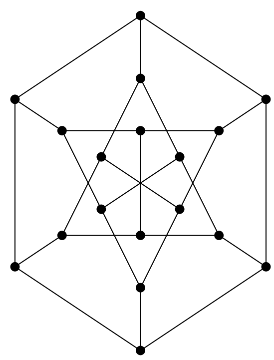
1. *Otázky* [vždy po **2 bodech**]
2. *Kódovací úkoly* [vždy po **4 bodech**]

Za 14 otázek a 2 kodovací je tedy možné získat až 36 bodů. Maximální zisk za úkol je ovšem **23 bodů**, můžete si tedy vybrat úlohy, které vás osloví. Odevzdejte v libovlně vhodném formátu včetně čitelných skenů, skriptů pythonu či jupyter notebooků. Vše zabalte do zip souboru s vhodnou strukturou a podmenováním souborů.

Otázka 1

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

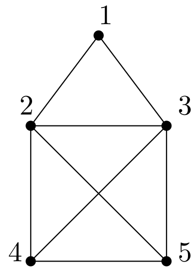
1. Cesta P_n
2. Kružnice C_n
3. Úplný graf K_n
4. Úplný bipartitní graf $K_{m,n}$
5. Papův graf ukázaný níže



Obrázek 1: Papův graf

Otázka 2

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



Obrázek 2: Graf H

Definujme *Laplacovu matici* L jako

$$L_{ij} = \begin{cases} \deg(v_i) & \text{pokud } i = j \\ -1 & \text{pokud } i \neq j \text{ a } \{v_i, v_j\} \in E(G) \\ 0 & \text{jinak} \end{cases}$$

Například pro ukázkový graf dostaneme

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

Mějme *k-regulární graf* $G = (V, E)$ velikosti $|V| = n$, tj. graf pro který víme, že $\deg(v_i) = k$ pro všechna $v_i \in V$. 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?

Otázka 3

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

Otázka 4

Mějme následující funkci

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

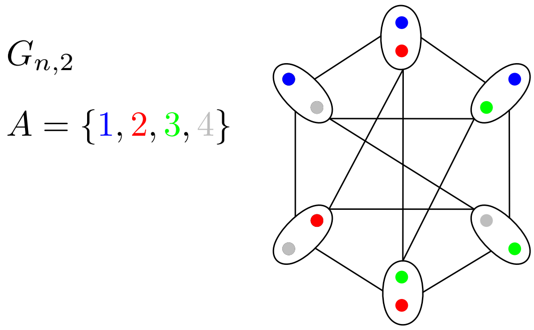
Vypočtěte gradient $\nabla f(x)$ a Hessian $\nabla^2 f(x)$ a rozhodněte (a zdůvodněte) zda-li je v bodě $(1, 1)^T$ splněna 1. podmínka pro lokální minimizátor (nulovost gradientu) či zda-li je splněna i 2. podmínka pro Hessovu matici.

Otázka 5

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

Otázka 6

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



Obrázek 3: Příklad grafu $G_{n,2}$

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.

Otázka 7

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

Otázka 8

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 $K_{m,n}$

Otázka 9

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 $\beta(G)$

Otázka 10

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).

Otázka 11

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.

Otázka 12

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.

$$f(x_1, x_2, x_3, x_4, x_5) = (x_1 \vee \overline{x_3}) \wedge (x_2 \vee x_3) \wedge (\overline{x}_3 \vee x_4) \wedge (\overline{x}_2 \vee \overline{x}_4) \wedge (x_4 \vee x_5) \wedge (\overline{x}_3 \vee x_5)$$

Otázka 13

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

$$f(x_1, x_2, x_3, x_4, x_5) = (x_2 \vee \overline{x_3} \vee x_4) \wedge (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee x_4) \wedge (x_1 \vee \overline{x}_2 \vee x_3 \vee \overline{x}_4) \wedge (x_4 \vee x_5)$$

Otázka 14

Na základě vysvětlení převodu 3-SAT na 3-COL nakreslete graf odpovídající následující formuli, ve kterém se následně testuje barevnost.

$$f(x_1, x_2, x_3, x_4, x_5) = (x_1 \vee x_2 \vee \overline{x_4}) \wedge (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_2 \vee \overline{x}_3 \vee x_4) \wedge (x_1 \vee \overline{x}_2 \vee x_3)$$

Kódovací úkol 1

Naprogramujte algoritmus pro testování isomorfismu grafů hrubou silou a otestujte to na pár příkladech grafů s využitím knihovní funkce pro testování isomorfismu. K tomu několik hintů. Nejpve schéma programu. To by mělo bez problémů otestovat následující:

```python
import networkx as nx
import numpy as np

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

# your code returning 'my_isom' boolean
my_isom = ...

# testing library
nx_isom = nx.is_isomorphic(G, nx.complement(H))

# check my code
nx_isom == my_isom
```

Lze předpokládat, že vrcholy grafu jsou reprezentovány jako pole čísel $[0, 1, 2, \ldots, n]$, což lze získat následujícím voláním

```python
>>> list(G.nodes)
        [0, 1, 2, 3, 4]
```

Navíc pomohou různé funkce pythonu na generování permutací a množin, viz

```python
>>> import itertools
>>> A = [0, 1, 2, 3, 4]
>>> list(itertools.permutations(A))        # všechny permutace
        [(0, 1, 2, 3, 4),
         (0, 1, 2, 4, 3),
         (0, 1, 3, 2, 4),
         ...
         (4, 3, 2, 0, 1),
         (4, 3, 2, 1, 0)]
>>> list(itertools.combinations(A,2))        # všechny páry
        [(0, 1),
         (0, 2),
         (0, 3),
         ...
         (2, 4),
         (3, 4)]
```

Kódovací úkol 2

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. Kód by měl vypadat následujícím způsobem.

```python
import networkx as nx
import numpy as np

G = nx.generators.small.small.petersen_graph()

# získejte největší nezávislou množinu hrubou silou
max_ind_set = ...

# Vytvořte algoritmus, který mírně podpořený využitím sousedství
max_ind_set = ...

# vhodně otestujte na knihovní funkci, že velikost vaší množiny odpovídá nezávislosti
```

Některé hinty, které by se při zpracování mohli hodit. Sousedy vrcholu získáváme normálně

```python
>>> G = nx.generators.small.petersen_graph()
>>> list(G.neighbors(0))
        [1, 4, 5]
```

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