MMAD - Úkol 2

Filip Ditrich

Unicorn University, Prague, Czech Republic
21. dubna 2024

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

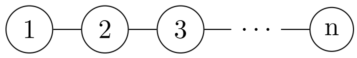
Řešení

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

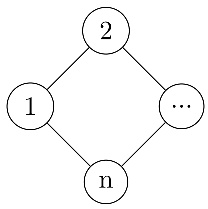
ODPOVĚĎ 1A

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.



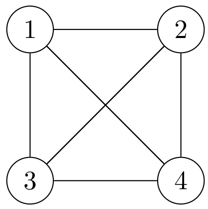
Obrázek (2) – Kružnice C_n

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.



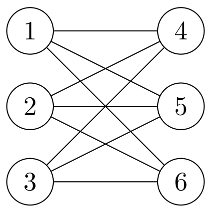
Obrázek (3) — Úplný graf K_4

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.



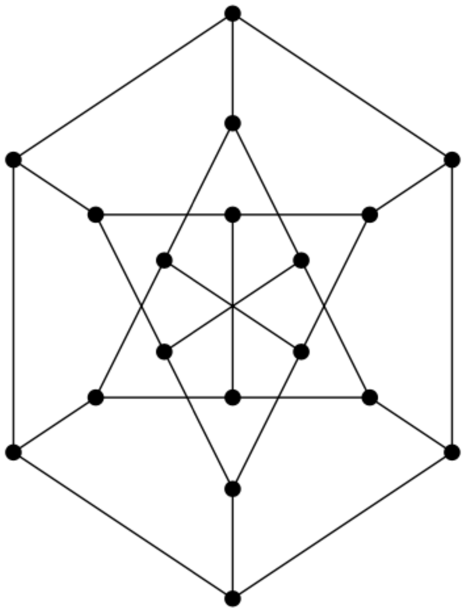
Obrázek (4) — Úplný bipartitní graf $K_{3,3}$

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)

TODO

Řešení

TODO

Úloha 3 (2 body)

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

Řešení

TODO

Úloha 4

(2 body)

TODO

Řešení

TODO

Úloha 5

(2 body)

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).

Řešení

TODO

Úloha 6 (2 body)

TODO

Řešení

TODO

Úloha 7 (2 body)

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

Řešení

TODO

Úloha 8

(2 body)

TODO

Řešení

TODO

Úloha 9 (2 body)

TODO

Řešení

TODO

Úloha 10 (2 body)

TODO

Řešení

TODO

Úloha 11 (2 body)

TODO

Řešení

TODO

Úloha 12 (2 body)

TODO

Řešení

TODO

Úloha 13 (2 body)

TODO

Řešení

TODO

Úloha 14 (2 body)

TODO

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)

TODO

Řešení

TODO