
PAL: 5. cvičení

21. 10. 2021

Př. 3/1: orientovaná kružnice

Každou hranu neorientované kružnice libovolně orientujeme. Jaký je vztah mezi počtem kořenů a listů v takto vzniklém grafu?

Př. 3/5: cesty délky 3

Popište, jak najdete a vypíšete všechny cesty délky 3 v acyklickém prostém grafu (bez násobných hran). Jaký je jejich maximální možný počet v závislosti na počtu uzlů grafu? Jaká bude asymptotická složitost Vašeho algoritmu?

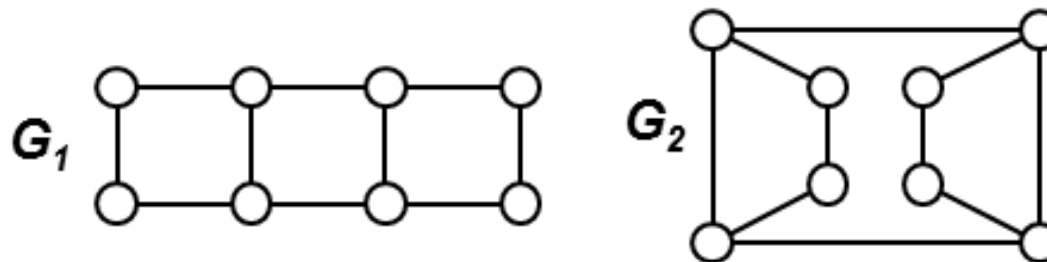
Izomorfizmy

Př. 5/1: izomorfizmy kružnice

Máme dvě neorientované kružnice stejné délky $k > 2$. Kolik mezi nimi existuje izomorfizmů?

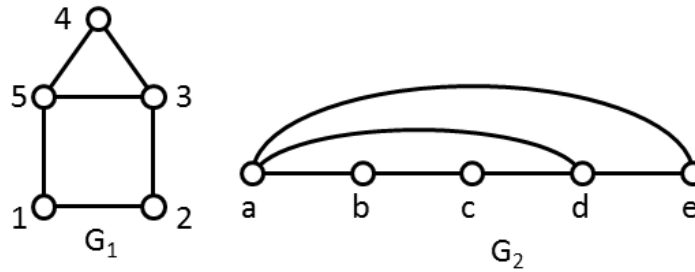
Př. 5/2: izomorfizmy grafů

Určete počet izomorfizmů mezi grafy G_1 a G_2 .



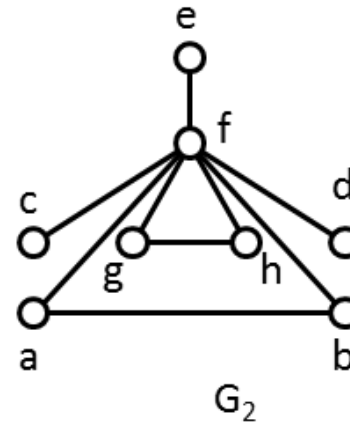
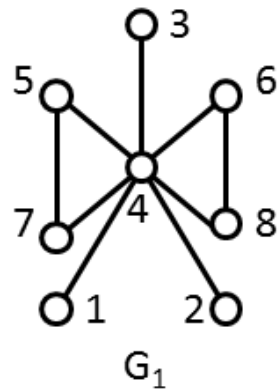
Př. 5/3a: počet bijekcí

Kolik je takových bijekcí mezi uzly grafů G_1 a G_2 na obrázku níže, které nejsou izomorfizmy?



Př. 5/3b: počet bijekcí

Kolik je takových bijekcí mezi uzly grafů G_1 a G_2 na obrázku níže, které nejsou izomorfizmy?



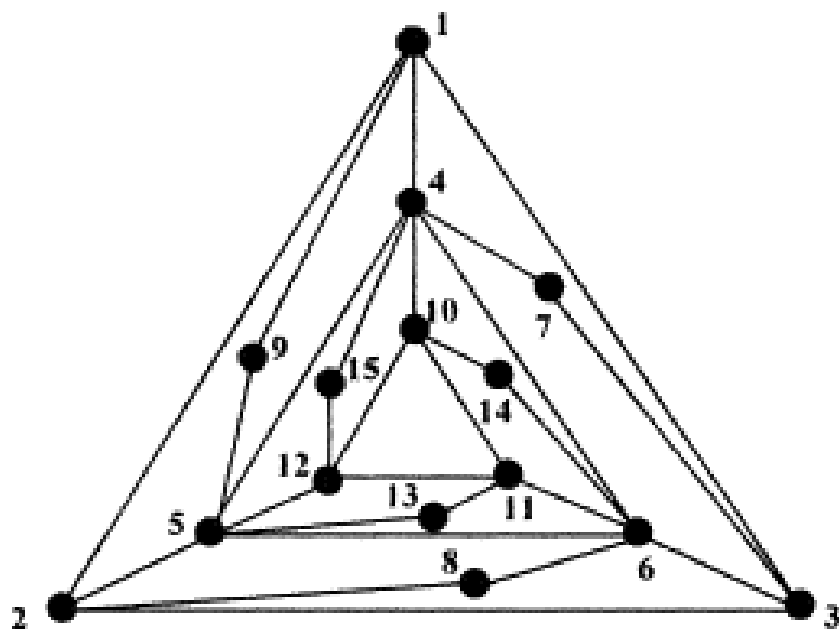
Př. 5/4: počet bijekcí

Máme dány dva neorientované grafy, každý obsahuje právě n uzlů a oba grafy mají skóre

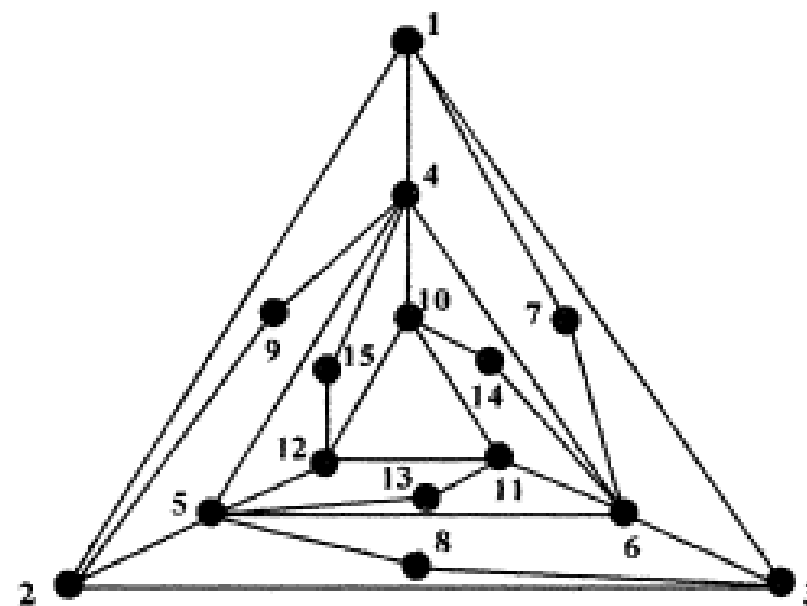
$(n - 1, n - 2, n - 3, n - 4, \dots, n/2 + 1, n/2, n/2, n/2 - 1, n/2 - 2, \dots, 3, 2, 1)$,
to jest skoro všechny uzly grafu mají navzájem různý stupeň, s výjimkou dvou uzlů, které mají stejný stupeň $n/2$. Jaká bude asymptotická složitost ověření izomorfizmu těchto dvou grafů v závislosti na hodnotě n ?

Př. 5/6: izomorfismus

Popište, jak budete co nejefektivněji rozhodovat, zda dva uvedené grafy jsou nebo nejsou izomorfní.



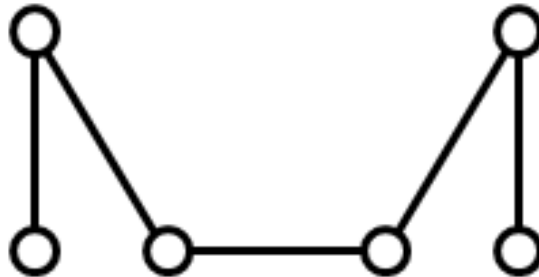
(a)



(b)

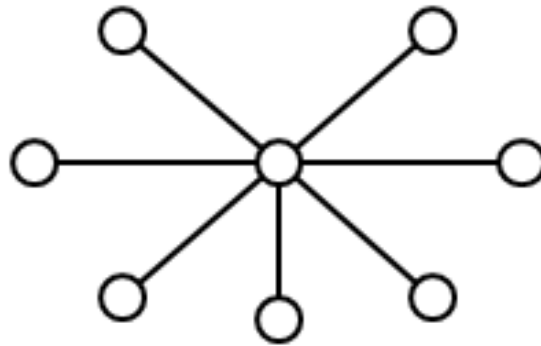
Př. 5/7a: tvorba certifikátu

Po sestavení certifikátu stromu odpovídá každému uzlu stromu určitý podřetězec konečného certifikátu. Sestavte certifikát daného stromu a určete, které jeho podřetězce odpovídají jednotlivým uzlům stromu.



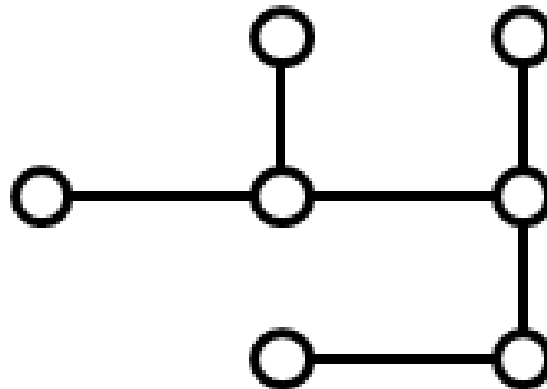
Př. 5/7b: tvorba certifikátu

Po sestavení certifikátu stromu odpovídá každému uzlu stromu určitý podřetězec konečného certifikátu. Sestavte certifikát daného stromu a určete, které jeho podřetězce odpovídají jednotlivým uzlům stromu.



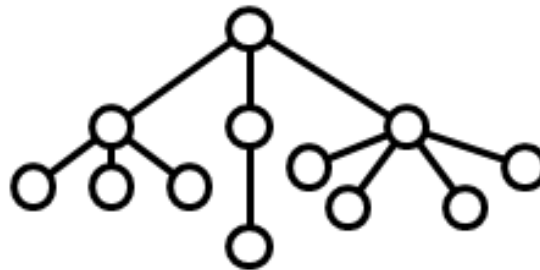
Př. 5/7c: tvorba certifikátu

Po sestavení certifikátu stromu odpovídá každému uzlu stromu určitý podřetězec konečného certifikátu. Sestavte certifikát daného stromu a určete, které jeho podřetězce odpovídají jednotlivým uzlům stromu.



Př. 5/7d*: tvorba certifikátu

Po sestavení certifikátu stromu odpovídá každému uzlu stromu určitý podřetězec konečného certifikátu. Sestavte certifikát daného stromu a určete, které jeho podřetězce odpovídají jednotlivým uzlům stromu.



Př. 5/8: rekonstrukce stromu z certifikátu

Rekonstruujte strom z certifikátu:

- a) 0101
- b) 0001010110010111
- c) 00010110010110010111
- d) 0000110001011110010001011111

Př. 5/9: listy v certifikátu

Je dán certifikát stromu. Vysvětlete, jak určíme počet listů tohoto stromu, aniž jej z certifikátu celý rekonstruujeme.

Př. 5/10*: maximální stupeň uzlu z certifikátu

Je dán certifikát stromu. Vysvětlete, jak určíme maximální stupeň uzlu tohoto stromu, aniž strom z certifikátu celý rekonstruujeme.

Př. 5/11: certifikát stromu s uzly stupně 1 a 3

Popište neformálně, jak bude vypadat certifikát stromu obsahujícího uzly pouze stupně 1 nebo 3. Navrhněte algoritmus, který pomocí certifikátu ověří, zda strom obsahuje uzly pouze stupně 1 nebo 3.

Kombinatorické algoritmy

Př. 6/1*: podmnožiny písmen

Je dána množina malých písmen $P = \{a, b, c, \dots, z\}$ bez diakritiky. Máme vygenerovat tabulku T s 8 sloupci, jejíž každý řádek bude obsahovat jednu 8-prvkovou podmnožinu množiny P tak, že každá buňka na řádku bude obsahovat právě jeden znak. Tabulka bude obsahovat všechny možné navzájem různé 8 prvkové podmnožiny P a žádná podmnožina se v T nebude opakovat. Určete, jak bude tabulka velká a zda ji váš počítač bude moci vyplnit během 1 sec.

Př. 6/2: všechny podmnožiny

Napište pseudokód funkce, která vypíše všechny neprázdné podmnožiny množiny $\{0, 1, 2, \dots, n - 1\}$.

Př. 6/3: přívětivé permutace

Mějme permutace množiny $M = \{1, 2, 3, \dots, n\}$, $n > 4$. Permutaci p této množiny prohlásíme za přívětivou, pokud platí:

- $p(3) \in \{3, n\}$,
- $p(n) \in \{3, n\}$,
- $p(1) = 1$,
- $p(2) = 2$,
- $p(i) \in \{4, \dots, n - 1\}$ pro $i = 4, \dots, n - 1$.

Určete počet přívětivých permutací množiny M .

Př. 6/4: Grayův kód

Předpokládejme, že každý prvek Grayova kódu G_n , jímž je n -tice nul a jedniček, bude uložen v poli znaků o délce n . Napište pseudokód rekurzivní funkce, která pro dané n vygeneruje a vypíše celý Grayův kód G_n .

Př. 6/5*: číslování permutací

Všechny permutace množiny M s 98 prvky očíslováme pořadovými čísly od 0 do $98! - 1$. V programu pak nepracujeme s permutacemi, ale jen s jejich pořadovými čísly. Víme, že budeme zkoumat pokaždé najednou 100 permutací, čili v paměti budeme muset mít uloženo právě 100 pořadových čísel různých permutací množiny M . Kolik minimálně bitů si musíme v paměti rezervovat, abychom si těchto 100 reprezentací mohli uložit?

Př. 6/6: předchozí podmnožina

Uvažujme všechny k -prvkové podmnožiny množiny

$M = \{1, 2, 3, \dots, n\}$, $1 \leq k \leq n$. Vyjděte z algoritmu transformujícího seznam prvků jedné podmnožiny na seznam prvků podmnožiny bezprostředně následující v lexikografickém uspořádání těchto podmnožin. Navrhněte a popište algoritmus, který bude transformovat seznam prvků jedné podmnožiny na seznam prvků podmnožiny bezprostředně předcházející v témže lexikografickém uspořádání. Bude mít stejnou asymptotickou složitost?

Př. 6/7: cykly permutací

Uvažujeme permutace množiny $M = \{1, 2, 3, \dots, n\}$. Cyklus délky k v permutaci p definujeme jako množinu $A = \{a_1, a_2, \dots, a_k\} \subset M$, pro kterou platí: $1 \leq a_1 < a_2 < \dots < a_k \leq n$, $p(a_j) = a_j + 1$ pro $1 \leq j < k$, $p(a_k) = a_1$. Určete, kolik je takových permutací množiny M , které obsahují právě dva cykly, z nichž jeden má délku 4 a druhý délku $n - 4$.

Př. 6/8*: pořadí permutace

Rank permutace π množiny $N = \{0, 1, 2, \dots, n - 1\}$ je pořadové číslo této permutace v seznamu všech permutací množiny N uspořádaném v rostoucím lexikografickém pořadí, přičemž prvky seznamu jsou číslovány od 0. Napište pseudokód funkce která v čase úměrném n vytiskne takovou permutaci π množiny N , jejíž rank je právě $n!/2$. Předpokládáme $n \geq 2$.