PAL: 6. cvičení

4. 11. 2021

Př. 3/5: cesty délky 3

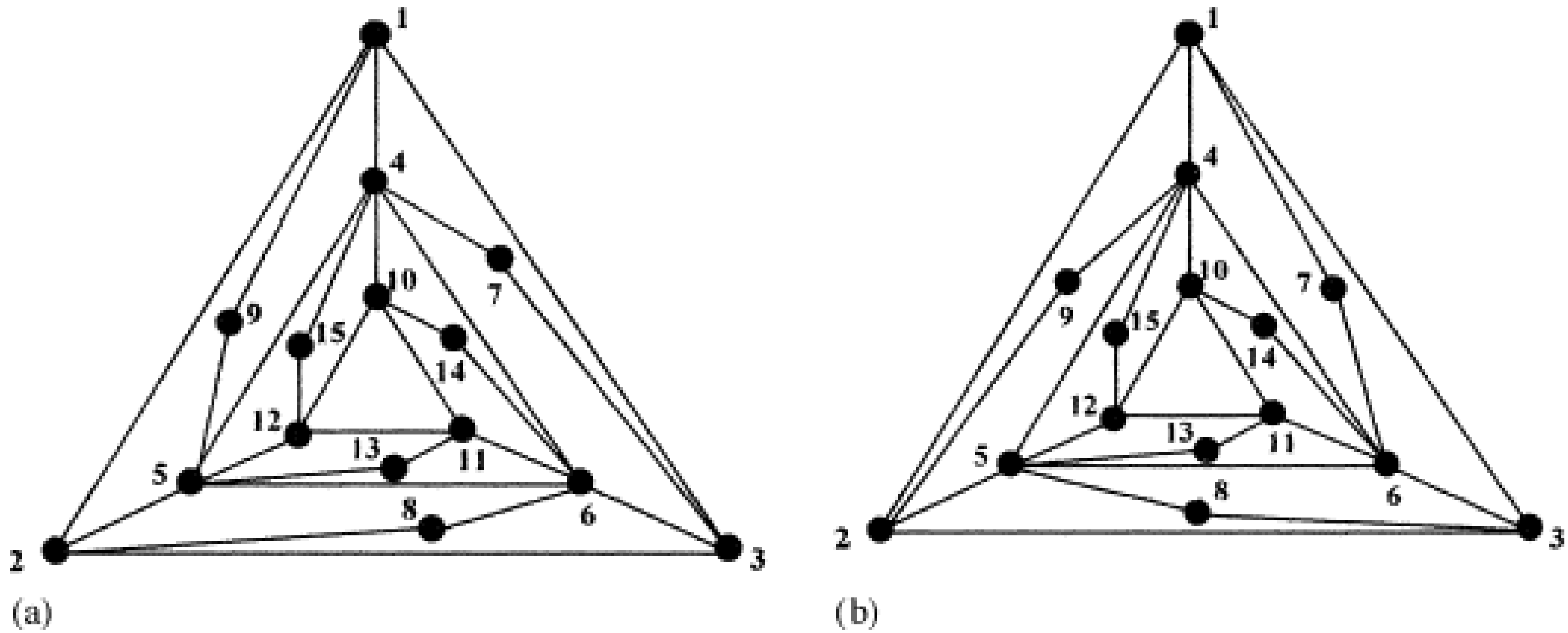
Popište, jak najdete a vypíšete všechny cesty délky 3 v acyklickém prostém grafu (bez násobných hran). Jaký je jejich maximální možný počet v závislosti na počtu uzlů grafu? Jaká bude asymptotická složitost Vašeho algoritmu?

Př. 5/4: počet bijekcí

Máme dány dva neorientované grafy, každý obsahuje právě n uzlů a oba grafy mají skóre
$(n-1, n-2, n-3, n-4, ..., n/2+1, n/2, n/2, n/2-1, n/2-2, ..., 3, 2, 1)$,
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

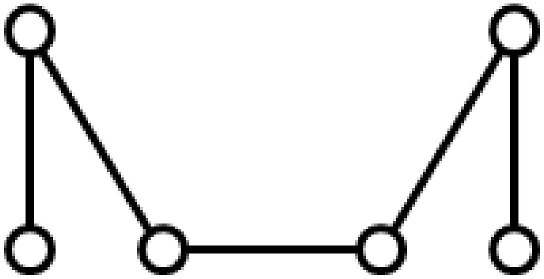
Popište, jak budete co nejefektivněji rozhodovat, zda dva uvedené grafy jsou nebo nejsou izomorfní.



Certifikáty stromů

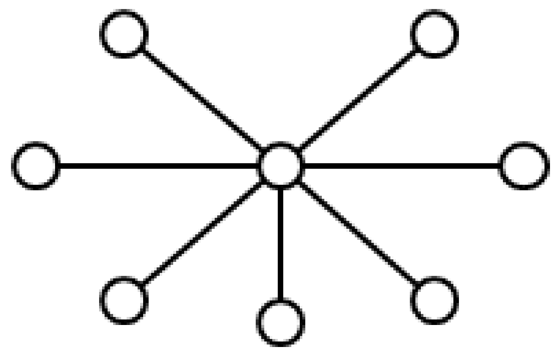
Př. 5/7a: tvorba certifikátu

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.



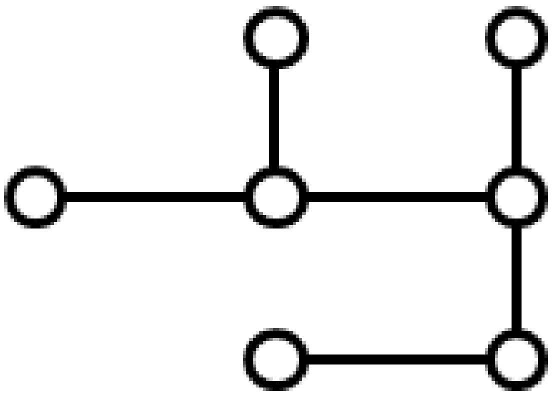
Př. 5/7b: tvorba certifikátu

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.



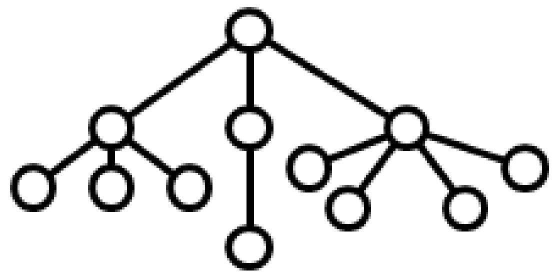
Př. 5/7c: tvorba certifikátu

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. 6/2: všechny podmnožiny

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

Př. 6/3: přívětivé permutace

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

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

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. 6/6: předchozí podmnožina

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

Př. 6/7: cykly permutací

Uvažujeme permutace množiny $M = \{1, 2, 3, \ldots, n\}$. Cyklus délky k v permutaci p definujeme jako množinu $A = \{a_1, a_2, ..., a_k\} \subset M$, pro kterou platí: $1 \leq a_1 < a_2 < \ldots < a_k \leq n$, $p(a_j) = a_j + 1$ 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 $N = \{0, 1, 2, \ldots, n-1\}$ 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 $n \geq 2$.

Konečné automaty, nedeterminizmus. Regulární výrazy

Př. 7/1: jazyky

Nad abecedou $\{0, 1\}$, jsou dány dva jazyky $L1$ a $L2$. Slova $L1$ jsou popsána regulárním výrazem $0 * 1 * 0 * 1 * 0*$, slova $L2$ jsou popsána regulárním výrazem $(01 + 10)*$.

a) Najděte nejkratší neprázdné slovo v průniku $L1 \cap L2$.

b) Najděte nejdelší slovo v průniku $L1 \cap L2$.

c) Najděte nejkratší slovo, které leží v L1, ale neleží v L2.

d) Najděte nejkratší slovo, které leží v L2, ale neleží v L1.

e) Najděte nejkratší slovo, které neleží v $L1 \cup L2$.

Př. 7/2: automaty

Nakreslete stavový diagram automatu přijímajícího právě všechna slova nad abecedou $\{0, 1\}$, která
a) obsahují podposloupnost 1010 alespoň jednou,
b) neobsahují podposloupnost 1010,
c) obsahují podposloupnost 1010 právě jednou,
d) obsahují podposloupnost 1010 nejvýše dvakrát.

Př. 7/3: regulární výrazy

Napište regulární výraz pro jazyk nad abecedou $\{0, 1\}$

a) jehož slova obsahují pouze nuly
b) jehož každé slovo obsahuje právě jedinou jedničku
c) jehož každé slovo obsahuje alespoň jednu jedničku
d) jehož každé slovo obsahuje alespoň dvě jedničky
e) jehož slova obsahují sudý počet jedniček
f) jehož slova obsahují lichý počet jedniček

Př. 7/4: nedeterministický automat: nuly a jedničky

Navrhněte NKA nad abecedou $\{0, 1, 2\}$, který v textu vyhledá všechny řetězce obsahující tři nuly a dvě jedničky.

Př. 7/5: nedeterministický automat: abcd

Navrhněte NKA nad abecedou $\{a, b, c, d\}$, který v textu vyhledá všechny řetězce ve tvaru $\#ba\#\#b\#$, kde symbol $\#$ představuje právě jeden libovolný znak z množiny $\{a, b, d\}$. Automat musí být schopen zpracovat celý text libovolné délky, tj. octnout se v koncovém stavu po přečtení posledního znaku každého výskytu hledaného řetězce.

Př. 7/6: automat z regulárního výrazu

Sestavte automat, který v textu nad abecedou $\{a, b, c\}$ vyhledává všechna slova popsaná regulárním výrazem $(ac * +bb) * a$.

Př. 7/7: uspořádaná slova

Mějme abecedu $A = \{a, b, c, ..., z\}$. Pořadové číslo znaku a bude 1, pořadové číslo znaku b bude 2, atd., až pořadové číslo znaku z bude 26. Slovo nad A nazveme uspořádané, pokud pro každý jeho znak platí, že všechny znaky za ním ve slově následující mají vyšší pořadové číslo než tento znak. Sestavte NKA, který vyhledá v textu nad abecedou A všechna uspořádaná slova.

Př. 7/8: řetězce se stejným počtem znaků

Sestavte NKA nad abecedou $\{0, 1, 2\}$, který v textu vyhledá všechny řetězce obsahující stejný počet znaků 0, 1 i 2.

Př. 7/9: certifikátový automat

Chtěli bychom sestavit konečný automat nad abecedou $\{0, 1\}$, který přijímá všechna slova, která představují certifikát nějakého neorientovaného stromu. Vysvětlete, zda to je či není možné, a pokud to možné je, popište, jak by se takový automat konstruoval.

Př. 7/10: rotovaný jazyk

Operace ROT zvolí některý znak x v řetězci a nahradí ho znakem v abecedě bezprostředně následujícím za x. Pokud x je poslední znak v abecedě, nahradí ho znakem prvním v abecedě. Sestavte NKA, který v textu vyhledá všechny podřetězce, které lze z daného vzorku $aabcb$ získat pomocí nejvýše dvou operací ROT. Abeceda je $\{a, b, c\}$.

Př. 7/11: regulární soupeři

Rozhodněte, zda regulární výrazy $(01 + 0) * 0$ a $0(10 + 0)*$ popisují stejný regulární jazyk.