PAL: 7. cvičení

3. 11. 2022

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.

```
cert[1..n] // certifikát stromu
i=1 // aktuální pozice v certifikátu
maxS = 0 // průběžně max. stupeň uzlu ve stromě

// Načte část certifikátu od daného uzlu dále od kořene
// (tj. 0...1 se stejným počtem 0 a 1).
// Vrátí stupeň vrcholu na aktuální pozici v certifikátu.
d = parsePart(level)
    d = 0
    i++ // přeskoč počáteční 0
    while cert[i]==0
        parsePart(level+1)
        d++
    end
    i++ // přeskoč 1 na konci
    maxS = max(maxS, d + (level>0))
end

d = dfs(0)
if i<n // strom s centrem velikosti 2
    maxS = max(maxS, d+1) // mezi 2 vrcholy centra je hrana
    d = dfs(0) // načtení druhé části stromu
    maxS = max(maxS, d+1) // mezi 2 vrcholy centra je hrana
end
```

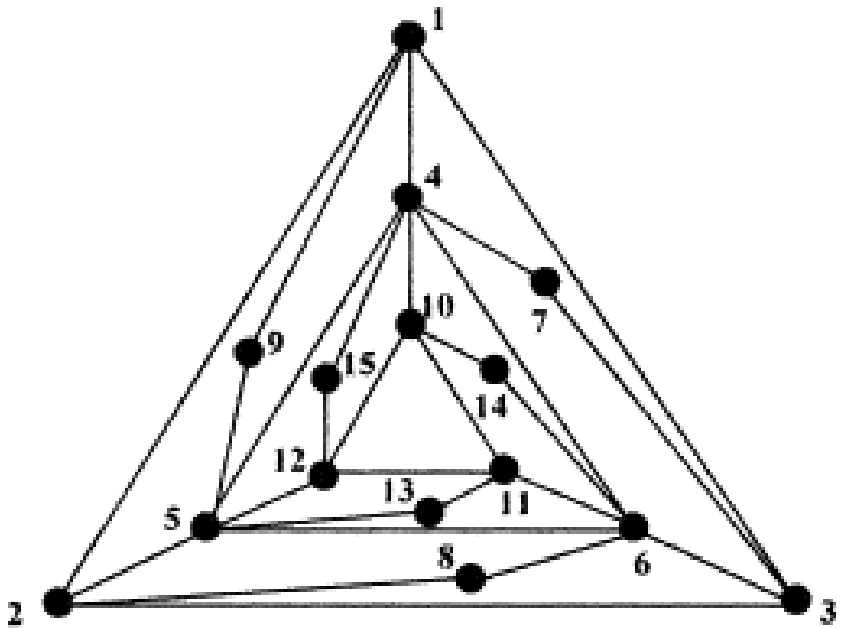
Př. 5/10*: maximální stupeň uzlu z certifikátu

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

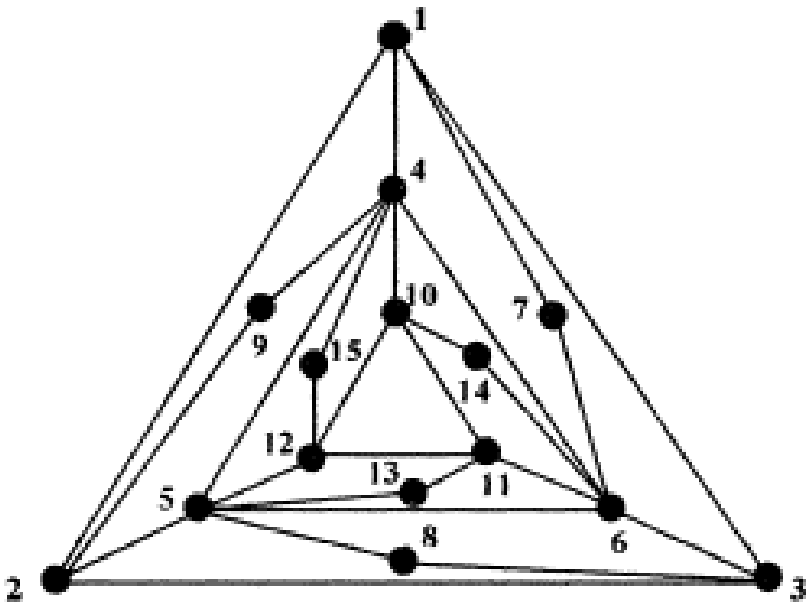
```
cert[1..n] // certifikát stromu
maxS = 0 // průběžně max. stupeň uzlu ve stromě
s = [] // zásobník
si = 0 // pozice na zásobníku

for i=1..n
    if cert[i]==0
        if si>0
            s[si]++
        end
        s[si++] = si>0 // 0 pro první úroveň, 1 pro další úrovně
    else
        maxS = max(maxS, s[si] + (si==1) && (i<n || leftSeen))
            // s[si] = počet synů aktuálního vrcholu
            // kód vpravo přidává hranu mezi vrcholy centra velikosti 2
        if si==1 && i<n
            // strom má centrum velikosti 2
            leftSeen = 1
        end
        si--
    end
end
```

Př. 5/6: izomorfismus

Zjistěte, zda dva uvedené grafy izomorfní. Popište, jak budete co nejefektivněji tento problém rozhodovat.



(a)



(b)

Kombinatorické algoritmy

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.