Zkouška IB002

29. května 2009

Jméno a příjmení: .....

UČO: .....

1

Do binárního vyhledávacího stromu (obecně nevyváženého), který je na počátku prázdný, jsou standardním algoritmem postupně přidávány položky s klíči $16, 4, 11, 9, 18, 13, 2, 6, 3, 19, 14, 15, 7, 1, 17, 12, 5, 10$ (v tomto pořadí).

(a) Nakreslete, jak vypadá tento strom po přidání všech položek. Tento první strom nevyvažujte.

(b) Pak obarvěte uzly tohoto stromu tak, aby se vzniklý strom stal červenočerným stromem. Černé uzly výrazně označte dvojitým kroužkem.

(c) Od tohoto okamžiku považujte strom za červenočerný. Přidejte do něho novou položku s klíčem 8 a proveďte rotace nutné k vyvážení stromu. Kolik rotací je potřeba? Nakreslete stavy mezi rotacemi a výsledný červenočerný strom.

2

Binární strom má n uzlů. Procházíme ho do šířky algoritmem *BFS* a začínáme v jeho kořeni.

(a) Jak se přitom *nejvýše* může zaplnit fronta použitá v algoritmu, v závislosti na n? Nakreslete, jak vypadá strom v tomto případě.

(b) Jak *nejméně* se tato fronta může zaplnit? Nakreslete, jak vypadá strom v tomto případě.

V obou případech vyjádřete velikost fronty asymptoticky, v závislosti na n.

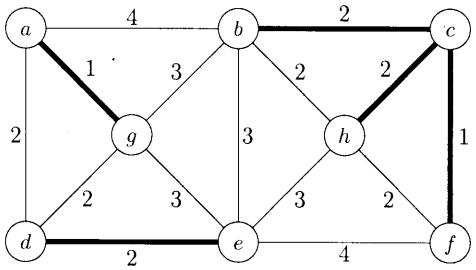
Zkouška IB002

29. května 2009

Jméno a příjmení: .....

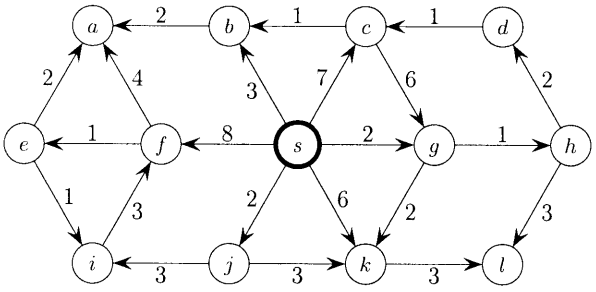
UČO: .....

3



Neorientovaný graf s vyznačenými hranami vyjadřuje stav konstrukce minimální kostry podle Kruskalova algoritmu po přidání pěti hran. Vyznačené hrany budou patřit minimální kostře. V dalších dvou krocích výpočtu budou přidány další dvě hrany kostry (šestá a sedmá), ale je více možností, kterými lze tyto dvě hrany vybrat. Napište všechny tyto možnosti, tj. všechny takové dvojice hran.

4



(a) Pro každý uzel grafu na obrázku určete jeho vzdálenost z počátečního uzlu s pomocí Dijkstrova algoritmu.

(b) Vyznačte v grafu ty hrany, které tvoří strom nejkratších cest z uzlu s.

Zkouška IB002

29. května 2009

Jméno a příjmení: .....

UČO: .....

5

Je dáno číselné pole A, indexované od 1 do n, určené k uložení zleva zarovnané minimové binární haldy velikosti n.

(a) Napište definici logické funkce (predikátu) `isHeap`, která zjistí, zda čísla v poli skutečně splňují podmínku minimové haldy. Funkce musí mít optimální časovou složitost.

(b) Určete časovou složitost této funkce.

(c) Zdůvodněte, proč je její složitost minimální.

Nápověda (c): hrany

6

Máme k dispozici základní operace binárního stromu, zejména: `isempty` (predikát prázdnosti), `left`, `right` (levý a pravý podstrom).
Binární strom je *preAVL*, když pro každý jeho uzel u platí, že délka nejdelší větve jdoucí z u doleva se od délky nejdelší větve jdoucí z u doprava liší nejvýše o 1. (Tedy vlastnost *preAVL* je tatáž jako *AVL*, jen nás zajímá pouze struktura stromu, nikoli hodnoty v jeho uzlech.)
Napište definici funkce `is_preAVL`, která otestuje, zda daný binární strom má tuto vlastnost. Nepředpokládejte žádnou konkrétní implementaci stromu, ale použijte jen uvedené metody.
Nápověda: pomocná funkce `avl_height` vrací *dvojici* hodnot, její první resp. druhou složku lze zpřístupnit projekčními funkcemi `fst`, `snd`.

```
is_preAVL(t)   = fst(avl_height(t))
avl_height(t)  = ...
```