



POSUDEK BAKALÁŘSKÉ PRÁCE

Název práce: Aplikace pro výuku stromů
Autor práce: Jakub Malíšek
Vedoucí práce: Mgr. Jiří Zecpal, Ph.D.

Splnila práce cíle uvedené v zadání? ano

Jazyková úroveň práce: dobrá

Formální zpracování: dobré

Komentáře a připomínky:

Místo názvu *diagram případu užití* na straně 10 má být *diagram případů užití*. V diagramu chybí jméno aktéra. Je v něm nic neříkající jméno *Actor*.

Textová část BP mi připadá obsahově nevyvážená. Teoretická část má jen 5 stran. Největší část BP tvoří programátorská dokumentace.

Poznámky k části aplikace označenou *Teorie*.

Definice pojmu kořen stromu je matoucí. Uvádí se v ní, že kořen je uzel, pro který platí, že z něho vede cesta ke každému uzlu. Uvedená skutečnost nijak nesouvisí s kořenem. Je to prostý důsledek faktu, že strom je souvislý graf.

Cesta k uzlu není jen posloupnost uzlů, jak se zde uvádí, ale patří do ní rovněž hrany, které jsou mezi těmito uzly.

Definice podstromu je chybná. Uvádí se zde, že *je to uzel, který přímo následuje za daným uzlem na cestě od kořene k listu*. Obecně podstrom není jen jeden uzel.

Vysvětlení, že *základní operace binárních vyhledávacích stromů jsou v nejhorším případě relativně rychlé* je vágní. Co se zde pojmem *relativně* myslí?

Zápis uzlů stromu je nejednotný. Někde se používá operátor indexu (*key[x]*), jinde se používá operátor tečka (*x.left*). Přehlednější by bylo zápis v celé teoretické části aplikace sjednotit. Nejlépe všude používat operátor tečka, který je přehlednější než operátor indexu.

V algoritmu 4 *průchodu do šířky* je použita proměnná *x*, která není nikde inicializována (není ji přiřazena žádná hodnota).

V algoritmu 6 *vyhledání uzlu s danou hodnotu* se opět používá proměnná *x*, která není inicializována.

Algoritmus 7 nalezení minima v závislosti na struktuře stromu buďto vrátí nedefinovanou hodnotu anebo se zacyklí.

Totéž platí pro *algoritmus 8 nalezení maxima*.

Pseudokód v *algoritmu 9 nalezení následníka* neodpovídá popisu algoritmu. V popisu se uvádí: *dokud nenarazíme na uzel, který je levým potomkem rodiče*. V pseudokódu se naopak srovnáním *x=y.right* zjevně zjišťuje, zda není pravým potomkem.

Obdobný rozpor je v *algoritmu 10 nalezení předchůdce*.

Definice faktoru vyvážení u AVL stromu je neúplná. Uvádí se zde, že je to *rozdíl výšek levého a pravého podstromu*. Odečítáme od výšky levého výšku pravého nebo naopak?

Schémat rotací u AVL stromů jsou poněkud matoucí. Na obrázcích 16, 17 a 19 se uvádí faktor vyvážení spodního uzlu roven 0. Přitom ve skutečnosti může mít libovolnou přípustnou hodnotu, tj. rovněž 1 nebo -1.

V popisu *algoritmu 13* rotace v AVL stromu se uvádí, že jako argument funkce se předává *odkaz na celý strom*. Co si pod pojmem odkaz na celý strom má uživatel představit?

Věta v části červeno-černých stromů „*Pokud uzel nemá potomka (jeho potomek je tím pádem list) nebo rodiče (uzel je v tom případě kořenem stromu), příslušné uložení má hodnotu NIL.*” je nesrozumitelná.

V *algoritmu 14* rotace v červeno-černém stromu se nikde nevyskytuje změna barev uzlů. U uživatele to může vyvolat mylný dojem, že barva žádného uzlu se při rotaci nezmění.

U vysvětlení B-stromu se někde používá pojem *prvek v uzlu*, jinde *klíč v uzlu*. Pro uživatele může být matoucí, proč se zde používají dva rozdílné pojmy.

Diplomant v definici B-stromu na jejím začátku uvádí, že mezi prvky je neklesající vztah $x.key_1 \leq x.key_2$, čímž se připouští vícenásobný výskyt stejných prvků. V pokračování definice se uvádí: „*Největší klíč v uzlu $x.child_1$ je pak menší než $x.key_1$, nejmenší klíč v $x.child_2$ bude větší než $x.key_1$ a největší klíč v $x.child_2$ bude menší než $x.key_2$ atd.*”, což naopak vylučuje možnost výskytu stejných prvků a je tak v rozporu s první částí definice.

Celkové hodnocení: C

Je práce vhodná pro propagační účely katedry? ne

V Olomouci dne 23. srpna 2015

RNDr. Arnošt Večerka

oponent