

POSUDEK NA DIPLOMOVOU PRÁCI VOJTĚCHA BARDIOVSKÉHO  
“IMPLEMENTATION OF OPERATIONS IN DOUBLE-ENDED HEAPS”

Pro práci s množinami, kde prvky mají danou preferenci, je výhodnější použít místo klasických datových struktur haldy, obzvláště, když se nemusí vyhledávat, zda prvek patří do množiny nebo nikoliv a úloha vyžaduje práci s prioritami prvků. Úlohy z praxe obvykle preferují vyhledávání prvků s malými prioritami (nebo s velkými prioritami) a to vedlo k tzv. jednokoncovým haldám, které se zaměřují na hledání a odstraňování prvků s nejmenší (nebo největší) prioritou. V poslední době se začaly zkoumat haldy, které umožňují hledání prvků jak s minimální, tak s maximální prioritou, tzv. dvoukoncové haldy. Byly navrženy obecné metody, jak z jednokoncových hald vytvořit dvoukoncové haldy. Jeden z cílů předložené práce byl právě navrhnout takovou efektivní metodu. Úlohy navíc obvykle vyžadují změny priorit. To jednokoncové haldy umí, ale pro dvoukoncové haldy se tyto operace nevyšetřovaly. Druhým cílem práce tedy bylo navrhnout metodu vytvoření dvoukoncové haldy, která by umožňovala efektivní provedení těchto operací. Lze říct, že autor tyto cíle splnil. Navrhl zobecnění tzv. ‘leaf correspondence’. Toto zobecnění umožňuje použití na širší spektrum hald, než to umožňovala leaf correspondence. Dále navrhl pro takto vytvořené dvoukoncové haldy algoritmy, které by zvyšovaly nebo zmenšovaly priority, a spočítal jejich složitost. Svou metodu ilustroval na třech důležitých konkrétních jednokoncových haldách – na Fibonacciho haldách, na rank-pairing haldách (na tyto haldy nejde použít leaf correspondence) a na thin haldách. Myslím, že to je důležitý příspěvek k současnému vyšetřování a vylepšování hald.

Bohužel musím napsat, že na práci je vidět, že autor neměl dost času, aby ji dostal do ideální podoby. Práce obsahuje některé nepřesnosti a měla by být přátelštější k čtenáři. Uvedu několik konkrétních připomínek.

Anglický a slovenský abstrakt se podstatně liší – anglický mluví o amortizované časové složitosti, slovenský pouze o časové složitosti (vypadlo tam slovo amortizované)

Na str. 5 je chybně popsána operace Delete-min, má tam být ‘least’ nikoliv ‘largest’. Popis operace Meld není formálně v pořádku, má být buď  $S_1 \cap S_2 = \emptyset$  nebo  $|S_1 \cap S_2| = 0$ , podmínka  $|S_1 \cap S_2| = \emptyset$  je sice jasná, ale formálně nesmyslná.

Na str 6. druhý odstavec – co znamená, že haldy jsou ekvivalentní?

Na str. 10 a později by min-heap měla být formálně definována. Je to halda, která splňuje min-order podmínku nebo abstraktní datový typ min-priority queue?

Na str. 12 v algoritmu Delete-min chybí  $Q_{max}Delete(z)$ . Takto by  $z$  zůstalo v max-haldě, kde by nemělo svůj pointer a bylo by i v bufferu. Na řádce 13 je lepší použít  $Q_{min}Delete - min$ , mohlo by to být jednodušší než  $Q_{min}Delete(y)$ .

Pro čtenáře by bylo dobré připomenout, že min-heap splňují min-order podmínku, a duálně max-heap, na tom je založena ‘leaf correspondence’.

Na str. 17 na řádce 39 programu podmínka není jednoznačná, čtenář musí zjišťovat, co autor chce napsat, mělo by to být vlídnější k čtenáři.

V odstavci za algoritmem je lepší používat přítomný čas (has) a ne minulý (had). Stejná poznámka platí pro str. 33.

Na str. 22 mi pořád není jasné, jestli je správně spočítána složitost, protože nevím, jaká je složitost INSERT. Abych se vyhnul případným potížím, nepoužíval bych operaci INSERT v Fix-L-node, ale přímo bych dával jednoprvkové stromy do haldy. Proti původnímu přístupu by se jen nedávaly do jednoho root list, ale do dvou.

Na str. 24 algoritmus nevypadá, že je korektní, protože prvek s minimální prioritou může být v bufferu. Tento případ není ošetřen.

Na 9. řádku zdola má být  $n$  vysazeno v matematice.

V operaci Decrease platí stejná poznámka s operací Insert jako pro Fix-L-node.

Věta za algoritmem vypadá nehezky a neanglicky.

Na str. 27 na prvním řádku má být jedno ‘that’ vynechané.

Na str. 32 by stačilo napsat, že Fibonacciho haldy splňují min-heap ordering podmínku, než ji tady popisovat.

Na str. 32 odhad, že počet stromů s různým rankem je nejvýše  $\log n$ , je chybný, má tam být buď  $O(\log n)$  nebo  $1.44 \log n$ .

Na str. 36 nerozumím tvrzení, že thin haldy jsou asymptoticky efektivnější než thick haldy, pokud si vzpomínám, tak mají všechny charakteristiky asymptoticky stejné (a jsou asymptoticky stejné jako pro Fibonacciho haldy).

Na str. 39 zápis algoritmu Fix-violating se mi nelíbí. Příkazy if na 15. a 20. řádku jsou matoucí (jsou po else, které zajišťuje, že podmínka je automaticky splněna).

Na str. 50 popis algoritmu Delete-min není v pořádku, tam má být změna pointeru na strom, jehož kořen má minimální prioritu, nikoliv změna pořadí. Stejná nepřesnost je na str. 51 v algoritmu Decrease.

Na str. 53 úvodní odstavec v části 3.3.2 neodpovídá dalšímu textu. Není dokázáno, že Fix-L-node bere konstantní čas, je ukázáno, že čas tohoto algoritmu je kompenzován úbytkem potenciálu.

Dvě obecné poznámky na konec. Není vhodné v textu používat dva různé způsoby pro zápis algoritmů. Čtenáři usnadní čtení jednotnost zápisů. Také není vhodné psát, že změna potenciálu je  $O(1)$ , když je záporná a kompenzuje spotřebovaný čas operace, který není  $O(1)$ . Změna se obvykle uvažuje v absolutní hodnotě a to čtenáře může mást. Asi by bylo lepší počítat přímo odhad na amortizovaný čas.

Závěr: Práce přináší zajímavé a netriviální výsledky a představuje cenný příspěvek do současné teorie hald. Je škoda, že autor neměl dost času, aby odstranil drobné nedostatky a práce proto není ideální. Přes tyto nedostatky myslím, že splňuje nároky na pěknou diplomovou práci a proto **doporučuji ji uznat jako diplomovou práci.**