TiRa labra - Toteutusdokumentti

Jarmo Isotalo

October 1, 2012

1 Toteutettavat algoritmit

Toteutan työssäni 3 kekoa, binäärikeon, binomikeon ja d-ary keon

2 Toteutuneet aika- ja tilavaativuudet (O-analyysi)

2.1 Aikavaatimus

Tarkastelen tässä vain muutaman eri tapauksen aikavaativuuksia:

1. Keon alustaminen - Create
   Jokaisessa keossa, binääri-,kolmi-, ja d-keossa operaatio on kutakuinkin saman kestoinen:
   (a) Aluksi alustetaan taulukko ja tallennetaan tietoon kunkin lapsien määrä. Binäärikeossa lapsia on kaksi, kolmikeossa kolme ja d-keossa d kappaletta. $O(1)$
   (b) Koska Create tehdään tyhjälle keolle, on operaatio vakioaikainen. Tässä asetetaan taulun ensimmäiseen indeksiin parametrina saatu arvo. $O(1)$
2. Kekoon lisääminen - Insert
   Jokaisessa keossa, binääri-,kolmi-, ja d-keossa operaatio on kutakuinkin saman kestoinen:
   (a) Aluksi parametrina saatu elementti lisätään keon viimeiseen indeksiin.$O(1)$
   (b) Sitten indeksille suoritetaan *heapify_up*, joka siirtää elementtiä ylöspäin, kunnes keko noudattaa taas kekoehtoa. Tässä oletetaan, että keko noudatti kekoehtoa ennen elementin lisäämistä. Tätä tapahtuu keon korkeuden verran. Eli insertin aikavaativuus on toteutuksessani $O(\log n)$

3. Keosta poistaminen - Delete
   Jokaisessa keossa, binääri-,kolmi-, ja d-keossa operaatio on kutakuinkin saman kestoinen:

   (a) Elementtiä keosta poistettaessa poistetaan elementti keon taulukon indeksistä 0. $O(1)$

   (b) Sen jälkeen siirretään keossa viimeisenä oleva elementti kekotaulukon indeksiin nolla. $O(1)$

   (c) Sitten kutsutaan *heapify_down* äsekettäin indeksiin nolla siirretylle, kunnes kekoehto toteutuu. $O(\log n)$

Kekojen toteutuken vuoksi O notaation ajat ovat samankaltaisia, mutta todellisuudessa lasten määrän lisääminen nopeuttaa keon toimintaa. Kunnolliset BenchMarkit tulossa TODO

	Binary Heap	Three Heap	D-ary Heap
Create	$O(1)$	$O(1)$	$O(1)$
Insert	$O(\log n)$	$O(\log n)$	$O(\log n)$
Delete	$O(\log n)$	$O(\log n)$	$O(\log n)$

2.2 Tilavaatimus

Tilavaativuuden arvoit eivät ole ihan hatusta heitettyjä, mutta ne riippuvat hyvin pitkälti toteutuksesta, joten rohkenen jättää ne melko avoimiksi. Ne toki tarkentuvat kun menille selviää kekojen ja niiden metodien toteutustapa

	Binary Heap	Binomial heap	D-ary
Insert	$O(\log n)$	$O(\log n)$	?
Delete	$O(\log n)$	$O(\log n)$	?
Build	$O(\log n)$	$O(\log n)$	$O(n)$

3 Lähteet

`https://en.wikipedia.org/wiki/Heap_(data_structure)`
Binäärikeko ja D-keko
`http://en.wikipedia.org/wiki/Binary_heap` `http://en.wikipedia.org/wiki/D-ary_heap` `http://www.cs.helsinki.fi/u/tapasane/keot.pdf` `http://www.cs.unc.edu/~plaisted/comp750/05-binheaps.ppt`
Binomikeko:
`http://cs.anu.edu.au/people/Warren.Armstrong/apac/trunk/module2/binomial_heaps.pdf` `https://www.cse.yorku.ca/~aaw/Sotirios/BinomialHeapAlgorithm.html` `http://en.wikipedia.org/wiki/Binomial_heap`