

Gegevensabstractie en -structuren

 tuyaux.winak.be/index.php/Gegevensabstractie_en_-structuren

Gegevensstructuren

Richting Informatica

Jaar 1BINF

Bespreking

Theorie

De theorie van dit vak is eigenlijk puur bedoeld om de praktijk tot een goed einde te brengen, dit vak wordt gedoceerd door Els Laenens en alhoewel soms wat traag, legt ze alles goed en duidelijk uit. Het leermateriaal bestaat uit een verzameling van slides en aangeraden is om het handboek te kopen. Dit handboek is niet echt nodig en zul je ook niet zo vaak gebruiken, maar kan wel nuttig zijn aangezien de meeste oplossingen van de praktijkopgaves hier letterlijk instaan (in C++ weliswaar). Het theorie-examen is niet echt moeilijk, enkele technieken een beetje vergelijken met elkaar.

De theorie wordt in de eerste helft van het semester gegeven waarna er meteen een examen van wordt gegeven. Vanaf dan wordt er enkel nog aan het project gewerkt.

Praktijk

Hier gaat het vak eigenlijk om. Enerzijds heb je gelijklopend met de theorielessen oefeningensessies die je wat handig moet maken in het gebruik van bepaalde algoritmen en dergelijke.

Na het theorie-examen in het eerste deel van het semester krijg je een groepsproject opgelegd dat relevant is aan de geziene theorie/oefeningen. Dit project bestaat enerzijds uit een aantal algoritmen/datastructuren die jullie moeten implementeren (rood-zwart boom bijvoorbeeld) en anderzijds uit een werkelijke opdracht. Deze opdracht valt al bij al nog wel mee als er goed in groep wordt gewerkt.

Tijdens de examenperiode zal je je project in groep moeten presenteren en onder andere delen van de code uitleggen.

Puntenverdeling

Praktijk: 14/20 (waarvan 10/20 op de oplossingen, 3/20 op de beoordelingen en 1/20 op de verantwoording). Theorie: 6/20.

Examenvragen

Academiejahr 2019 - 2020 (2de zittijd)

Individueel Project

Hieronder vind je de input bestanden voor je individueel geïmplementeerde datastructuur.

23.txt

234.txt

bst.txt

heap.txt

hm-lin.txt

hm-quad.txt

hm-sep.txt

linked.txt

queue.txt

rb.txt

stack.txt

Theorie

1.

1. Vul de tabel in.
2. Hoe bekom je de uitkomst bij Bubble Sort? Toon aan via pseudocode.
3. Geef de definitie van algoritme A is $O(f(n))$.

	Average case	Worst case
heapsort		
treesort		
quick sort		
merge sort		
insertion sort		
selection sort		
bubble sort		

1. 2 segmenten pseudocode (gelijkaardig aan 2de zit 2018-2019), Zijn volgende algoritmes correct?
2. Gegeven een 2-3 boom en een 2-3-4 boom, beiden hebben dezelfde hoogte en hetzelfde aantal nodes alsook dezelfde elementen. Is het algoritme als we inserten/deleten bij beide bomen hetzelfde? Verklaar.
3. Bewijs dat de inorder traversal op een binaire boom alle elementen in volgorde teruggeeft.
4. Leg collisions uit bij hashing, leg in een aantal zinnen de oplossing uit en toon aan welke oplossing het meest relevant is bij extern geheugen.
5. Leg externe mergesort uit aan de hand van duidelijke figuren.

Oefeningen

1. Doe een heapsort op een gegeven array (in array vorm en stop na 2 deletes).
2. Gegeven inserts en deletes en maak hier een 2-3 boom van.
3. Gegeven inserts en deletes en maak hier een 2-3-4 boom van.
4. Gegeven inserts en deletes en maak hier een rood-zwart boom van.
5. Gegeven een binaire zoekboom. Voer hier preorder traversal op uit.

Academiejaar 2019 - 2020 (1ste zittijd)

Individueel Project

Hieronder vind je de input bestanden voor je individueel geïmplementeerde datastructuur.

23.txt

234.txt

bst.txt

heap.txt

hm-lin.txt

hm-quad.txt

hm-sep.txt

linked.txt

queue.txt

rb.txt

README.txt

run.sh

stack.txt

system.txt

todot.sh

Theorie

Deel 1

Meerkeuzevragen met giscorrectie +1 voor elk juist antwoord, -0,4 voor elk fout antwoord

1. The ADT _____ is value-oriented
 1. binary tree
 2. list
 3. stack
 4. queue
 5. sorted list
2. An ADT's _____ govern(s) what its operations are and what they do.
 1. data structure
 2. specifications
 3. implementation
 4. documentation
3. Which of the following is NOT true about converting infix expressions to postfix expressions?
 1. the operators always stay in the same order with respect to one another
 2. the operands always stay in the same order with respect to one another
 3. an operator will move only "to the right" with respect to the operands
 4. all parentheses are removed
4. Each merge step of the mergesort requires _____ major operations.
 1. $4 \cdot n - 1$
 2. $(n - 1) / 2$
 3. $3 \cdot n - 1$
 4. $n - 1$
5. A dummy head node _____.
 1. facilitates adding nodes at the end of the linked chain
 2. is used to store the first item in the linked chain
 3. is the second node in the linked chain
 4. is always present, even when the linked chain is empty
6. Which of the following operations leaves a queue unchanged?
 1. enqueue
 2. dequeue
 3. createQueue
 4. getFront
7. Which of the following is true about all doubly linked chains
 1. each node has both a precede pointer and a next pointer
 2. the precede pointer of the last node has the value NULL
 3. the last node points to the first node
 4. the precede pointer of the first node points to the last node

8. In the binary search tree implementation of the ADT table, the maximum number of comparisons required by the tableInsert operation is equal to ____.
1. the number of nodes in the binary search tree
 2. the height of the binary search tree
 3. the number of leaves in the binary search tree
 4. the number of internal nodes in the binary search tree
9. The traversal of a binary tree is ____.
1. $O(n)$
 2. $O(1)$
 3. $O(n^2n^2)$
 4. $O(\log^2 n \log^2 n)$
 5. In the worst case, a binary search tree is ____.
 6. $O(\log^2 n \log^2 n)$
 7. $O(n)$
 8. $O(1)$
 9. $O(n^2n^2)$

Deel 2

Open vragen: schrijf je antwoorden op deze opgavebladen (hoeft niet in dezelfde volgorde)

1.

1. Vul in de volgende tabel de overeenkomstige groeigraadfuncties in.
2. Bewijs dat je antwoord op meerkeuzevraag 4 correct is.
3. Geef de definitie van Algoritme A is $O(f(n))$

	Average case	Worst case
heapsort		
treesort		
quick sort		
merge sort		
insertion sort		
selection sort		
bubble sort		

1. Wat weet je van het aantal knopen in een binaire boom met hoogte h ? Bewijs! Geef volgende aspecten duidelijk weer als je in je bewijs inductie gebruikt:
 1. wat is de waarde waarop je de inductie toepast?
 2. geef en bewijs het basisgeval
 3. geef de inductiehypothese (IH)
 4. geef en bewijs het inductiegeval
 5. waar gebruik je de inductiehypothese (IH) in het bewijs?

Als je een lemma gebruikt, geef dat dan zonder bewijs.

1.
 1. Illustreer aan de hand van een duidelijke figuur wat een indexfile is.
 2. Wat zijn de 3 belangrijkste voordelen van het gebruik van een indexfile?
2. Geef 2 verschillen tussen een gerichte graaf en een niet-gerichte graaf.

Succes!!!

Academiejaar 2018 - 2019 (2de zittijd)

Theorie

1. Wat is de efficiëntie van het binary-search algoritme en bewijs dit.
2. Geef de pseudocode van selection sort en leid hieruit af wat de efficiëntie is.
3. Bij de torens van Hanoi met moves(n), hoeveel moves gebeuren er met n schijven en bewijs dit via inductie.
4. Welke output krijgen we bij deze code.

```
Writeback(string:s)
```

```
Write("enter writeback with string:s")
```

```
If s is empty
```

```
    Do nothing
```

```
Else
```

```
    Writeback(s-eersteCharacter)
```

```
    Write(s)
```

```
Write(leaving writeback with s)
```

1. Zijn volgende algorithmes correct

```
Print1:
```

```
    For i = 0 to getsize
```

```
        Print(getitem(i))
```

```
Print2:
```

```
Current = l.head
```

```
    While current not null
```

```
        Print(l.getitem)
```

```
        Current = current->next
```

1. Wat zijn waarde en positie adts en geef 3 voorbeelden van elks.
2. Welke 3 implementaties zijn er van adt queues en wat zijn hun voor en nadelen?
3. Welke implementaties zijn er van grafen. Geef voor elk een voorbeeld.
4. Leg aan de hand van een tekening uit hoe externe hashing met index files werkt

Oefeningen

1. Maak een 23-boom door deze gegevens toe te voegen en te verwijderen
insert(1), insert(11), insert(3), insert(9), insert(6), delete(1), insert(7), insert(8),
insert(4), delete(9), insert(10), insert(2), delete(3)
2. Maak een rood-zwart-boom door deze gegevens toe te voegen en te verwijderen
Zelfde gegevens als hierboven
3. Leg treesort uit met een 234 boom en volgende gegevens.
3, 9, 8, 2, 6, 5, 1, 10
4. Leg uit: Hashmap met hash functie $\text{key mod } 11$ en separate chaining, maar gebruik een binaire zoekboom in plaats van gelinkte ketting.
5. Maak een heap door deze gegevens toe te voegen en te verwijderen
insert(60), insert(70), insert(90), insert(120), delete(120), insert(10), insert(80),
insert(30), insert(0), insert(50), delete(90).
6. Als we een array ombouwen tot een heap volgens het heapSort algoritme, wordt volgende for-lus uitgevoerd:

```
for(index=n-1 down to 0)
    heapRebuild(anArray, index, n)
```

Kan deze for-lus ingekort worden? Motiveer je antwoord.

Academiejaar 2018 - 2019 (1ste zittijd)

Theorie

Deel 1

Meerkeuzevragen met giscorrectie +1 voor elk juist antwoord, -0.4 voor elk fout antwoord

1. The specifications of an ADT's operations indicate _____.
 1. what the operations do
 2. how to implement the operations
 3. how to store the data in the ADT
 4. how to carry out the operations
2. Data structures are part of an ADT's _____.
 1. definition
 2. implementation
 3. specifications
 4. usage

3. Which of the following is a base case for a recursive binary search algorithm? (first is the index of the first item in the array, last is the index of the last item in the array, and mid is the midpoint of the array).
1. $\text{last} > \text{first}$
 2. $\text{first} > \text{last}$
 3. $0 \leq \text{first}$
 4. $\text{last} \leq \text{SIZE}-1$
4. The towers of Hanoi problem makes exactly _____ moves when it starts with N disks.
1. $2N^2N$
 2. $2N-12N-1$
 3. N^2N^2
 4. N^3-1N^3-1
5. Which of the following is the postfix form of the infix expression: $(a+b)-c/d$
1. $a\ b\ +\ c\ d\ /\ -$
 2. $a\ b\ +\ c\ -\ d\ /\$
 3. $a\ +\ b\ -\ c\ /\ d$
 4. $a\ b\ +\ c\ d\ -\ /\$
6. The _____ operation is the most significant difference between the ADT priority queue and the ADT table
1. createPQueue
 2. pqIsEmpty
 3. pqInsert
 4. pqDelete
7. A graph is _____ if each pair of distinct vertices has a path between them.
1. complete
 2. disconnected
 3. connected
 4. full
8. Which of the following is NOT a required step for a deletion operation under external hashing of an index file?
1. search the index file for the index record
 2. search the data file for the block of the data record
 3. delete the index record from the index file
 4. delete the data record from the data file
9. In a B-tree of degree 5, each node other than the root contains _____ records.
1. between 2 and 4
 2. between 2 and 5
 3. between 3 and 4
 4. between 3 and 5

10. The hashing of an index file is an appropriate implementation for which of the following operations on an external table?
1. sorted traversal
 2. retrieval of the smallest or largest item
 3. insertion of an item
 4. range queries that require ordered data

Deel 2

Open vragen

1.
 1. Vul in de volgende tabel de overeenkomstige groeigraadfuncties in.
 2. Geef voor elk van deze functies de intuïtieve betekenis en de praktische gevolgen.
 3. Geef de definitie van Algoritme A is $O(f(n))$.

	Average case	Worst case
heapsort		
treesort		
radix sort		
quick sort		
merge sort		
insertion sort		
selection sort		
bubble sort		

1. Vul in en bewijs per inductie: Een binaire boom van hoogte h heeft precies $2^h - 1$ knopen. Geef volgende aspecten duidelijk weer in het bewijs:
 1. basisgeval
 2. inductiehypothese in alle detail
 3. inductiegeval
 4. het gebruik van de inductiehypothese
2. Vul in volgende zin de correcte groeigraadfunctie in. Het doorlopen van een pad van de wortel naar een blad in een 2-3 boom is $O(\dots)$. Bewijs dit in zo veel mogelijk detail dat dit zo is!

Oefeningen

1. Begin met een lege (max-)heap en voer onderstaande operaties uit. insert(40); insert(30); insert(10); insert(60); delete(); insert(90); insert(20); insert(70); insert(100); insert(50); delete()
 1. Geef enkel de resulterende boom
 2. Is deze boom gebalanceerd/vol/compleet? Geef telkens aan waarom wel/niet?
2. Begin met een lege tabel, voer er onderstaande operaties op uit, en geef enkel de resulterende boom weer als de tabel geïmplementeerd wordt met een 2-3-4 boom. insert(2); insert(20); insert(6); insert(16); insert(10); delete(2); insert(12); insert(14); insert(8); delete(16); insert(18); insert(4); insert(9); insert(2)
3. Begin met een lege tabel, voer er onderstaande operaties op uit, en geef enkel de resulterende boom weer als de tabel geïmplementeerd wordt met een rood-zwart boom (stel rode pointers voor met een stippellijn en zwarte met een volle lijn) insert(2); insert(20); insert(6); insert(16); insert(10); delete(2); insert(12); insert(14); insert(8); delete(16); insert(18); insert(4); insert(9); insert(2)
4. Stel dat we in volgende array $a=[11, x, 33, 15, 22, 4]$ (m.a.w. $a[0]=11$, $a[1]=\text{leeg}, \dots$) een element met searchkey 2 toevoegen. We gebruiken quadratic probing met $h(\text{key}) = \text{key} \bmod 6$.
 1. Geef de probe sequence.
 2. Worden alle items van de array bereikt?
 3. Welke wel/niet?
 4. Hoe komt dit?
5. Geef de pseudocode voor een preorder traversel in een 2-3-boom die als volgt geïmplementeerd wordt:

```
class TwoThreeTree:
    subtrees = [None, None, None] # zullen elks een instance zijn van TwoThreeTree
    items = [] # zullen integers zijn waarbij None nooit voorkomt, max lengte is 2
    function preorder(f):
        """
        Doe een preorder traversal en pas telkens functie f toe.
        :param f: functie die werkt op integers, niet op nodes of trees
        """
```

Academiejahr 2015 - 2016

Theorie

1. Leg uit: ADT, gegevensstructuren en geef het verband ertussen
2. Geef alle mogelijke implementaties van een table
3. Geef een theorema van binaire bomen dat te maken heeft met aantal knopen en de hoogte van een boom. Geef het inductiebewijs
4. 10 meerkeuze vragen vanuit de testbanken.

Oefeningen

1. Vergelijk alle cases van complexiteit tussen een circulaire gelinkte ketting en een binaire zoekboom voor het zoeken van een item met een searchkey
2. Insertion en deletion bij een binaire boom. Geeft de resulterende boom, de hoogte en preorder traversal
3. Hashtable (met grootte 6) met $h(x) = X \bmod 6$, insert met quadratic probing items: 10,17,22,28
4. Insertion en deletion bij table met 2-3-4-boom implementatie : geef resulterende 2-3-4 boom
5. Insertion en deletion bij table met rood-zwart-boom implementatie met zelfde instructies als vorige vraag: geef resulterende rood-zwart boom
6. Heapsort op array (letters A tot J; random volgorde) tot 2 letters gesorteerd zijn. Tussenschappen (in vorm van array) geven.

Academiejaar 2014 - 2015

Theorie

Eerst waren er 10 multiple choicevragen (met giscorrectie) die bijna letterlijk uit de testbanken kwamen.

Daarna waren er open vragen:

1. Leg adhv een tekening uit hoe het toevoegen van elementen aan een 2-3-boom ervoor zorgt dat de boom gebalanceerd blijft.
2. Welke positie-georiënteerde ADT's ken je?
3. Hoeveel knopen heeft een volle binaire boom met hoogte h en bewijs deze formule.

Academiejaar 2013 - 2014 - 1e zittijd

Theorie

1. Bewijs theorema over binaire bomen. cfr. Aantalnodes in volle binaire bomen = $2^h - 1$
2. 4 goede vragen voor recursieve oplossing.
3. Voordelen van een indexbestand.

Praktijk

1. Kan vorm heap veranderen door het verwijderen en hertoevoegen van een item?
2. START: Lege binaire zoekboom. Insert(D), ~(C), ~(A), ~(F), del(C), insert(I), insert(B), ~(G), ~(J), ~(E), del(D). Geef de resulterende boom, de hoogte en preorder traversal.
3. Heapsort array CGFBEDAH (2 items sorteren)
4. Lege tabel, voer deze operaties uit: ins(1), ~(10), ~(3), ~(8), ~(5), del(1), ins(6), ins(7), ~(4), del(8), ins(9), ~(2), del(7) op:
 1. 2-3-4 boom
 2. rood-zwart boom

(ins = table.tableInsert(), del = table.tableInsert(), ~ = hetzelfde qua operatie vorig item.

1. Resulterende hash table (lengte 6) $h(x)=x \bmod 6$, quadratic probing, insert 10, 17, 22, 28.

Academiejaar 2011 - 2012 - 2de zittijd

Theorie

1. Je krijgt een tekening waar een programma LANGS de wall of ADT operations gaat en niet VIA (violating the wall) zoals figuren 3-9 op p143 in het boek en je moet zo gedetailleerd mogelijk uitleggen wat je ziet op die tekening.
2. Wat is het voordeel van een Dummyheadknoop in een gelinkte lijst?
3.
 1. Geef de formele definitie van een 'Algoritme A is $O(f(n))$ '.
 2. Vergelijk de efficiëntie van de bewerkingen (doorlopen, toevoegen, verwijderen, zoeken) van een binaire zoekboom met die van een 2-3boom.
4. Geef de verschillende soorten hashing.
5.
 1. Geef een informaticaprobleem dat je het best kan oplossen mbv multiple indexing.
 2. Leg uit waarom je vindt dat multiple indexing de beste oplossing is.
 3. Maak een tekening van je gekozen gegevensstructuur.

Praktijk

1. Schrijf de pseudocode voor een recursief algoritme dat voor een letter alle permutaties van a tot die letter uitprint.
2. Maak een binaire zoekboom in aangegeven volgorde. Vertrek telkens van een lege binaire zoekboom.
 1. W T N J E B A
 2. W T N A B E J
 3. A B W J N T E
 4. Verwijder J uit de laatste boom.
3. Gegeven een heap in array representatie die op 1 of meerdere plaatsen niet klopt.
 1. Schrijf een recursief algoritme heapify() die deze heap corrigeert (de functie gaat er vanuit dat de 2 deelbomen wel correcte heaps zijn).
 2. Schrijf stap voor stap in boomrepresentatie uit hoe de heap mbv je functie gecorrigeerd wordt.
4. Gegeven 2 arrays: [8 6 4 2] en [2 4 6 8].
Gebruik insertion en selection sort om beide arrays te sorteren, geef bij elke 'swap' de nieuwe staat van de array weer.
5. Zet de rood-zwartboom uit het handboek (oef 10 H12 p718) op in een 2-3-4 boom.

Academiejaar 2010 - 2011 - 1ste zittijd

Theorie

1. Bij de implementatie van het type hashtable kies je best een tablesizedie priem is. Waarom?
2. Stel een tabel op van alle geziene (zoek)algoritmen. Vermeld hierin hun worst case en average case efficiëncy.
 1. Bespreek de verschillende graden van efficiëntie intuïtief.
 2. Schrijf pseudocode voor één van de algoritmes die je boven hebt vermeld en beschrijf hoe je aan de gekomen efficiëntie komt.
3. Wat is het aantal knopen voor een volle binaire boom op hoogte h ?
 1. Wat is het totaal aantal knopen in een binaire boom van hoogte h ?
 2. Bewijs je bevindingen.
4. Geef de formele definitie van:
 1. Een algemene zoekboom
 2. Een B-boom van graad m
 3. Wanneer is m optimaal?
 4. Schrijf pseudocode voor een tableretrieve.

Praktijk

1. Afbeelding Missing | Geef steeds de structuur van de boom weer.
 1. Voeg toe (in deze volgorde): 75,20,16,48,1.
 2. Delete: 70.
1. Afbeelding Missing | Gegeven de volgende graaf:
 1. Doorloop de boom BFS en geef de bezochte volgorde.
 2. Geef de BFS spanning tree.
 3. Geef de minimum spanning tree.
1. Zet volgende array om in een heap d.m.v. heaprebuild.
 1. 14 20 3 16 24 9 12 8 18
1. Is dit een correcte Red-Black tree? Afbeelding Missing:
 1. Indien ja, voeg 18 toe aan de boom en geef de resulterende boom weer.
 2. Indien nee, leg uit waarom.

Labotoets

1. Vertrek van je reeds geschreven ADT binaire zoekboom. De unieke zoeksleutels zijn die van het type INTEGER. Voeg er de volgende operaties aan toe:
 1. Geef het item met de grootste zoeksleutel weer.
 2. Geef het item met de kleinste zoeksleutel weer.
 3. Geef de gemiddelde waarde van de zoeksleutels.

2. Voer (als test) volgende reeks operaties uit:
 1. Creëer een lege binaire zoekboom.
 2. Voeg achtereenvolgens 12,18,14,6,4,9,20,17,5,7,15 toe.
 3. Bepaal de hoogte van de boom.
 4. Geef de boom weer, in order.
 5. Geef het item met de kleinste sleutel.
 6. Geef het item met de grootste sleutel.
 7. Geef de gemiddelde waarde van de sleutels.
 8. Voeg achtereenvolgens 2,3,19 toe aan de boom en verwijder 7,20,6,12.Herhaal stap c -> g.
3. Implementeer het ADT minheap, gebruik makende van het ADT binaire zoekboom. De elementen opgeslagen in de heap zijn uniek en hebben een zoeksleutel van het type INTEGER. We hebben de volgende operaties nodig.
 1. Create
 2. Wis een minheap
 3. Bepaal of een minheap leeg is.
 4. Voeg een nieuw element toe aan de minheap.
 5. Verwijder het element met de kleinste zoeksleutel.
4. Voer de volgende testoperaties uit.
 1. Creëer een lege minheap.
 2. Voeg items met zoeksleutels 12,6,16,21,14,2,18,20,17 toe.
 3. Verwijder 3 keer achter elkaar het item met de kleinste zoeksleutel.
5. Pas het ADT minheap aan, zodat de zoeksleutels niet uniek hoeven te zijn. In dit geval wordt bij het verwijderen van een item het item verwijderd dat zich het langst in de heap bevindt. Schrijf hier zelf een testprogramma voor.

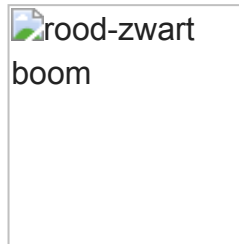
Academiejahr 2009 - 2010 - 1ste zittijd

Theorie

1. Bij double hashing geldt dat elke zoekreeks alle locaties in de hashtable bezoekt als de tabelgrootte en stapgrootte relatief priem zijn, geef een concreet voorbeeld dat aantoont dat niet noodzakelijk alle locaties bezocht worden als tabelgrootte en stapgrootte niet relatief priem zijn.
2. Waartoe dient de efficiëntie-analyse van algorithmen? Waarop is deze gebaseerd bij interne methoden en bij externe methoden. Motiveer!
3. Stel het ADT tabel wordt geïmplementeerd met een boomstructuur, hoeveel elementen zou je dan plaatsen in een knoop indien (bespreek je keuzes)
 1. de boom volledig in intern geheugen zit.
 2. de boom volledig in extern geheugen wordt opgeslagen omdat deze te groot is.
4. Leg a.d.h.v. een figuur uit hoe je een index bestand kan gebruiken bij hashing. Geef een opsomming van de opeenvolgende stappen die nodig zijn bij het verwijderen van een element uit deze structuur.

Praktijk

1. Beschouw onderstaande rood-zwart boom.



1. Voeg aan de boom achtereenvolgens de elementen met zoek sleutels 64, 25, 84, 52, 47 toe en geef telkens resulterende rood-zwart boom weer.
2. Verwijder van de boom bekomen door vorige operaties het element met zoek sleutel 30, en geef de rood-zwart boom weer.
2. Geef een concreet voorbeeld van een 2-3 boom met hoogte 4 waarbij door het verwijderen van één enkel element de hoogte 3 wordt, zorg er bovendien ook voor dat de boom waar je mee vertrekt zoveel mogelijk elementen bevat en dat het element dat je verwijdert niet in een blad zit.
3. Veronderstel ADT Stack. Schrijf in pseudo-code een operatie voor dit ADT waarbij elk voorkomen van een gespecificeerd item uit de stack verwijderd wordt, terwijl de volgorde van de andere items gerespecteerd blijft. Maak hiervoor enkel gebruik van de ADT operaties.

Labotoets

1. Vertrek van je reeds geschreven ADT binaire zoekboom. De unieke zoek sleutels zijn van het type INTEGER. Voeg er de volgende recursieve operaties aan toe:
 1. Geef het item met de grootste zoek sleutel weer.
 2. Bepaal het hoogste niveau van de boom dat vol is (dat met andere woorden het maximale aantal knopen bevat).
2. Voer volgende reeks operaties uit:
 1. Creëer een lege binaire zoekboom.
 2. Voeg achtereenvolgens 6,8,15,3,9,7,14,1,10,4 en 13 toe.
 3. Verwijder hieruit achtereenvolgens 14,8 en 6.
 4. Voeg hieraan 11 en 12 toe.
 5. Bepaal de hoogte van de binaire zoekboom.
 6. Geef het element met de grootste zoek sleutel weer.
 7. Geef weer tot welk niveau de boom vol is.
3. Implementeer het ADT tabel, d.m.v. hashing. De unieke zoek sleutels zijn van het type INTEGER. Gebruik een hashtable met tableSize 7. Botsingen (collisions) worden opgevangen d.m.v. separate chaining. In plaats van enkelvoudige gelinkte lijsten, worden er binaire zoekbomen gebruikt. Voorzie ook een operatie waarmee je kan weergeven welke items er in je hashtable aanwezig zijn (met hun juiste locatie), zodat je je hashtable kan testen.

4. Voer volgende reeks operaties uit:
 1. Creëer een lege tabel (tableSize 7)
 2. Voeg toe: 30, 51, 65, 40, 28, 7, 26, 16, 49, 70, 12, 37, 35.
 3. Toon de hashtabel.
 4. Verwijder hieruit achtereenvolgens 7, 26, 49, 30
 5. Voeg toe: 72, 17, 36, 104, 87
 6. Toon de hashtabel
5. (Enkel voor wie tijd over heeft) zorg er voor dat je hashtabel hiervoor geïmplementeerd dynamisch kan groeien. Van zodra één van de zoekbomen een hoogte groter dan " krijgt, moet de tableSize van de hashtabel aangepast worden: de nieuwe tableSize wordt het eerste priemgetal groter dan 2 keer de oude tableSize. Je mag als maximale grootte van je hashtabel 29 aannemen.

Academiejaar 2007 - 2008 - 1ste zittijd

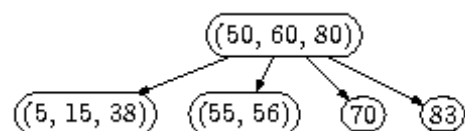
Theorie

1. Geef de formele definitie van $O(f(n))$. Bespreek $O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(n^3)$ en $O(2^n)$ op een intuïtieve manier.
2. Gegeven de verzameling van alle verplichte vakken in de bacheloropleiding, met hun noodzakelijke voorkennis als een lijst van vakken die afgelegd dienen te worden. Je kan pas voor een vak inschrijven als je alle vakken van de voorkennis afgelegd hebt. Je wenst na te gaan in welke volgorde je de verschillende vakken kan volgen. Leg uit welke toepassing van graven je hiervoor gezien hebt en hoe je te werkt gaat.
3. Wat is timedriven Simulatie & eventdriven simulatie
4. Gegeven B-boom met graad 5 en hoogte 3. Bepaal de maximum en minimum aantal records van de boom en min & max aantal knopen van de boom

Oefeningen

1. Schrijf een recursieve procedure voor het kleinste getal te zoeken van een BST (Binary Search Tree) (Pseudo code)
2. Schrijf een procedure die het aantal items van een queue telt zonder de queue te verwijderen (Pseudo code)
3. Insert 75 , 20 , 16 en 18 in de 2-3-4 boom (Fig. fig:234tree)

Zet deze 2-3-4 boom om in een Red Black tree na het inserten van deze getallen.



Academiejaar 2002 - 2003 - 1ste zittijd

Theorie

1. Geef definitie en motivatie van ADT.
2. Geef formele definitie van 2-3boom.
3. Leg het hoe en waarom uit van het indexeren van externe bestanden (mbv figuren).
4. Bespreek het optreden en behandelen van botsingen bij hashing.

Oefeningen

1. Schrijf in pseudocode een operatie voor het ADT Queue, waarbij het aantal items van een Queue geteld wordt. De Queue moet blijven bestaan, maak hierbij enkel gebruik van ADT operaties.
2. Veronderstel een binaire boom waarvan de items Integers zijn. Schrijf in pseudocode een recursieve operatie die de som van de elementen van de binaire boom berekent.
3. Oefening op gegeven B-tree (orde?, insert en delete).

Academiejaar 2000 - 2001 - 1ste zittijd

Theorie

1. Bespreek hashing en collision handling. Voor welk ADT wordt hashing gebruikt? Is hashing geschikt voor alle soorten bewerkingen op dit ADT? Waarom (niet)?
2. In welk opzicht zijn 2-3 zoekbomen beter dan binaire zoekbomen voor de implementatie van een tabel? Zijn bomen die knooppunten toelaten met meer dan 3 kinderen dan nog beter? Motiveer je antwoord.
3. Wat verstaan we onder het indexeren van een externe file en waarom wordt het gebruikt? Bespreek 2 manieren waarop index files kunnen georganiseerd worden.

Praktijk

1. Bij de pointer gebaseerde implementatie van het ADT stack willen we een operatie implementeren die een lege stack aanmaakt. Welke van de 4 volgende stukjes code geeft hiervoor de beste implementatie weer? Verklaar je antwoord.
 1. `PROCEDURE Create1 (s: stack); begin s := NIL end;`
 2. `PROCEDURE Create2 (s: stack); begin NEW(s); s := NIL end;`
 3. `PROCEDURE Create3 (VAR s: stack); begin s := NIL end;`
 4. `PROCEDURE Create4 (VAR s: stack); begin NEW(s); s := NIL end;`
2. Sorteert de volgende rij getallen met heapsort: 44 55 12 42 94 18 6 67. Laat bij elke stap de heap zien, zodat de werking van het algoritme duidelijk blijkt.

Academiejaar 1997 - 1998 - 1ste zittijd

Theorie

1. Bespreek de implementatie van een Heap. Waarom is deze implementatie juist zo geschikt voor Heaps? Waarvoor worden Heaps gebruikt?
2. Wat bedoelt men met collision handling? Bespreek de verschillende mogelijkheden.
3. Waarvoor is een indexrecord of indexfile nuttig? Bespreek de voordelen.

Oefeningen

1. Heapsort de volgende sequentie: 25 57 48 37 12 92 86 33.
2. Stel je hebt een priority queue geïmplementeerd door binaire boom. Geef de voorstelling van de priority queue na elk van de volgende operaties (links = priority queue, midden = prioriteit, rechts = inhoud): Create(pq), Insert(pq,2,6), Insert(pq,2,7), Insert(pq,3,6), Insert(pq,1,8), Insert(pq,3,3), Remove(pq), Insert(pq,4,2), Insert(pq,6,2), Remove(pq), Remove(pq).

Academiejaar 1995 - 1996 - 1ste zittijd

1. Geef een overzicht van de verschillende datastructuren en bespreek telkens de mogelijke implementaties.

Academiejaar 1994 - 1995 - 1ste zittijd

1. Bespreek het algoritme dat de infix expressie evalueert. Van welke ADT(s) maak je gebruik?
2. Geef de definities van tabellen en bomen. Verband?
3. Schrijf een programma voor "topological sorting". Gebruik recursie.
4. Toon aan dat de delete operatie van 2-3bomen correct werkt.

Academiejaar 1994 - 1995 - 2de zittijd

1. Definitie: ADT + leg uit: position oriented en value oriented + voorbeelden.
2. Definitie heap.
3. Leg uit: Hashing.

Oud examenvoorbeeld

1. Definieer het ADT Stack. Is een procedure StackTop(S,e): kopieer topelement van S met
 - preconditie: S bestaat
 - postconditie: S onveranderd, e = envereist? Hoe is de vergelijkbare implementatie bij Queue's?
2. Definieer binaire bomen. Bespreek de verschillende implementatiemogelijkheden. (Geef telkens type declaraties geassocieerd met de voornaamste datastructuur)