Schriftelijk tentamen Datastructuren
Woe 5 jan 2011 14-17 uur
Met uitwerkingen

1.a. Geef een compacte definitie van wat er bij Datastructuren verstaan wordt onder een Abstract Data Type (ADT).
b. Werk op papier een sorteer ADT uit die, met behulp van zo weinig mogelijk stapel ADTs, een invoer van losse integers gesorteerd uitvoert (elke stapel heeft 1 I/O-veld)
(8 punten)

Antwoord op 1a ter waarde van max 4 punten:
Onder een Abstract Data Type wordt verstaan de specificatie van een datastructuur voor een bepaald doel (b.v. Binaire Zoek Boom) met duidelijk omschreven gebruiks mogelijkheden (b.v. opzoeken, invoegen, weglaten), eisen waaraan de invoer moet voldoen (datatypes, precondities b.v. array van integers); eigenschappen waaraan de uitvoer voldoet (postcondities b.v. gesorteerd array) en welke externe (voor de gebruiker beschikbare) en interne (voor de gebruiker afgeschermde) functionaliteit geboden wordt en hiervoor nodig is (b.v. sorteerfuncties) zodat een gebruiker de datastructuur kan benutten door deze te voorzien van de juiste invoer en aan te roepen met de juiste parameters qua functionaliteit.

Antwoord op b ter waarde van max 4 punten:
Met behulp van 3 stapel ADT's (Stapel 1,2 en 3) met bijbehorend IO-veld kan een rij losse integers als volgt gesorteerd worden:
Stapeli heeft IO-veldi; stapels leeg bij aanvang
Initialisatie: Zet invoer rij integers in Stapel1 m.b.v. push
Herhaal tot Stapel 1 en 2 leeg:
Haal integers m.b.v. pop van Stapel1(dan wel 2) in IO-veld1(dan wel 2) en bewaar hoogst langsgekomen waarde van {IO-veld1/2 dan wel IO-veld3} in IO-veld3 push de andere waarde op Stapel2 (dan wel 1); als Stapel1 (dan wel2) leeg, push IO-veld3 op Stapel3 en wissel rol Stapel 1 en 2 om.

Eindafhandeling: totdat Stapel3 leeg: pop van Stapel3 volgende element en voer die uit (naar b.v. invoer array, zodat deze na afloop de gesorteerde waardes bevat)

2. Geef een Array Mapping Functie die de matrix M[i][j] voor opslag van 16bit integer waarden met i ε [3..8] en j ε [16..25] afbeeldt op een lineair stel bytes beginnend met adres 40.
(6 punten)

Antwoord op 2 goed voor max 6 punten:
M[i,j] i ε[3..8], jε[16.25] waarvan elk element 2 bytes(16bits) in beslag neemt, wordt als volgt afgebeeld op een 1-dimensionaal byte array beginnend op adres 40:
Deze 6*10=60 elementen die 120 bytes in beslag nemen komen afhankelijk van of rij voor rij dan wel kolom voor kolom wordt doorlopen terecht op adres:

{(i-3)*10+(j-16)}.2+40 of {(j-16).6+(i-3)}.2+40 Hiermee M[3,16]->40 en M[8,25]->158 en nemen zo precies 120 bytes [40..159] in.

3. Geef voor 4 uitvoeringen van een lineaire lijst, te weten ongesorteerd (L1), gesorteerd (L2), circulair verbonden met next pointer (L3) en dubbel circulair verbonden met next en previous pointer (L4) aan wat de complexiteit is bij elk van deze uitvoeringen voor de volgende elementaire operaties bij aanwezigheid van de hulpvariabelen Aantal, Current:
- insert van 1 element
- delete van 1 element
- find next (sorted) element
- find previous (sorted) element
-insert after current
- insert before current
(12 punten)

Antwoord op 3 goed voor max 12 punten (2 voor elke kolom of 3 voor elke rij): (findnext en findprev in sortering)
L1 en L2 zijn array's; L3 en L4 lijst van punten verbonden middels 1 of 2 pointers

Lijst	insert	delete	findnext	findprev	insaftercur	insbeforecur
L1	O(1)	O(N/2)	O(N/2)	O(N/2)	O(1)	O(1)
L2	O(logN)+N/2	O(logN)+N/2	O(1)	O(1)	O(1)+N/2	O(1)+N/2
L3	O(N/2)	O(N/2)	O(1)	O(N/2)	O(1)	O(N/2)
L4	O(N/2)	O(N/2)	O(1)	O(1)	O(1)	O(1)

4. Bij dit tentamen is apart een ADT van een binaire zoek boom gevoegd; deze ADT is opgezet met als elementair datatype een knoop met twee pointers naar linker- en rechter-kind.
Geef aan hoe de ADT functies vereenvoudigd kunnen worden als het elementair datatype niet alleen twee pointers naar z'n kinderen heeft, maar tevens een pointer naar de ouder.
(7 punten)

5. a. Geef aan wanneer een Binaire Zoek Boom met uniek voorkomende integers voor het opzoeken van een bepaalde integer een complexiteit slechter dan O(logN) heeft.
b. Geef alle mogelijke zoekboom realisaties voor de 3 knopen 14, 18 en 25 als die at random worden toegevoegd.
c. Welke van de onder b gegeven BST realisaties hebben een min pad vorm ?
(8 punten)

Antwoord op 5a goed voor max 3 punten:
Een BST hoeft in het geheel niet goed gebalanceerd te zijn na 1^{e} vulling; zelfs een backbone structuur is mogelijk (als b.v. de waardes gesorteerd worden toegevoegd); opzoekcomplexiteit is daarom op z'n best O(logN) en op z'n slechtst O(N).

Antwoord op 5b goed voor max 3 punten:
Qua invoegvolgorde zijn met 3 knopen 3! Mogelijkheden:
14,18,25 geeft backbone structuur naar rechts vanaf wortel met 14
14,25,18 14 in wortel, 25 en 18 rechts, 18 linkerkind van 25
18,14,25 18 in wortel, 14 linkerkind, 25 rechterkind; compleet
18,25,14 18 in wortel, 14 linkerkind, 25 rechterkind; compleet
25,14,18 25 in wortel, 14 en 18 links, 18 rechterkind van 14
25,18,14 geeft backbone structuur naar links vanaf wortel met 25

Antwoord op 5c goed voor max 2 punten:
Aleen de invoegvolgordes 18,14,25 en 18,25,14 geven de minpadvorm

6. a.Geef aan hoe in een threaded BST de pointers naar de L en R kind knopen gebruikt worden als er geen kind aanwezig is.
b. Hoe onderscheid je pointers naar kinderen van pointers naar opvolgers?
(6 punten)

Antwoord op 6a goed voor max 3 punten:
Bij een threaded BST wordt in principe het ontbrekende kind link verlegd naar die knoop die in de InOrder doorloop (LOR volgorde) de volgende te bezoeken knoop zou zijn; voor het ontbrekende linkerkind is dit een link naar de Ouderknoop, voor het ontbrekende rechterkind de volgende knoop die bezocht zou worden bij de InOrder doorloop.

Antwoord op 6b goed voor max 3 punten:
Om pointers naar een kinderknoop te onderscheiden van pointers naar een opvolgerknoop (bij een lege kindknoop) is minimaal 1 bit nodig; soms wordt het tekenbit hiervoor gebruikt (de adresseringsrange wordt daarmee wel gehalveerd).

7. a. wat verstaan we onder de (hoogte)balans van een BST?
gegeven is een BST met 7 knopen:
b. Tussen welke grenzen kan de (hoogte) balans van deze BST varieren?
c. Geef met een tegenvoorbeeld aan dat de aanwezigheid van blad(eren) op 1 nivo niet garandeert dat de BST zo goed mogelijk hoogte gebalanceerd is.
(8 punten)

Antwoord op 7a goed voor max 3 punten:
Onder de hoogtebalans verstaan we het verschil in niveaus onder een knoop in de boom.

Antwoord op 7b goed voor max 3 punten:
Bij een BST van 7 knopen, waarvoor een perfect gebalanceerde BST mogelijk is met overal hoogtebalans 0 waarden maar ook een backbone structuur met 7 niveaus waar de hoogtebalans varieert tussen 1 en 6, afhankelijk van of op het 1 na laatste niveau of vanuit de wortel de balans wordt opgemaakt.

Antwoord op 7c goed voor max 2 punten:
c. Een overtuigend voorbeeld is de backbone structuur die maar 1 blad heeft en maximaal ongebalanceerd is.

8. Hoe onderscheidt een MaxHeap representatie zich van een BST?
(5 punten)

Antwoord op 8 goed voor max 6 punten:
Bij een BST geeft een InOrder doorloop een gesorteerde rij knoopwaarden;
Bij een MaxHeap geldt alleen dat langs elk simpel pad van wortel naar blad de knoopwaardes aflopend gesorteerd zijn (m.a.w. de maximale waarde staat gegarandeerd in de wortelknoop.

9. Voor het snel toegankelijk houden van informatie binnen een volume (3D) kan gebruik gemaakt worden van een Binary Space Partitioning Tree (BSP) of een Octree.
Geef voor deze concurrerende volume opdelingen aan wat de opdeling regelt en wat het essentiële verschil tussen beide aanpakken is.
(7 punten)

Antwoord op 9 goed voor max 8 punten:
Een BSP tree gebruikt de waarden van de op te bergen data zelf om de ruimte (herhaald) te halveren in 1 van de 3 richtingen bij toerbeurt, terwijl een octtree een voorgedefinieerd hiërarchisch grid oplegt aan de data, waarbij alleen de diepte van opdeling in elke hiërarchische cel afhangt van de aangeboden data.

10. a. Geef de definitie van een minimale perfecte hash functie.

Gegeven de volgende reeks 29 studentID's:

0601977
0535729
0326658
0305715
0517062
0333816
0364460
0404659
0607932
0613088
9901647
0409871
0656860
0622184
0611735
0430935
0441317
9921192
0518492
0345296
0433373
0650064
0650617
0525839
0632864
0639877
0744735
0616990
0524034

b. maak de statistiek op van het voorkomen van elk digit

c. welke digits komen het meest in aanmerking voor opname in een hash key gebaseerd op een 2 digit selectie?

d. hoeveel botsingen komen er nog voor als je de twee beste digits gebruikt om een hash key ε [00..99] te construeren?

e. zijn eventuele botsingen met een simpele lineaire rehash op te lossen?

(13 punten)

Antwoord op 10a goed voor max 3 punten:

Een minimale, perfecte hash functie berekent een unieke index in een lijst die precies even groot is als het aantal onder te brengen datawaarden.

Antwoord op 10b goed voor max 3 punten:
Tabel van frequenties digitwaardes tegen digitposities:

	d1	d2	d3	d4	d5	d6	d7
0:	27	0	6	3	4	0	3
1:	0	0	5	5	2	4	1
2:	0	0	5	2	1	1	4
3:	0	5	6	3	2	6	1
4:	0	5	3	4	2	1	4
5:	0	5	3	4	0	2	4
6:	0	11	1	3	4	5	2
7:	0	1	0	2	4	4	5
8:	0	0	0	1	6	2	2
9:	2	2	0	2	4	4	3
0-9:	29	29	29	29	29	29	29 (ter controle)

Antwoord op 10c goed voor max 2 punten:
Kies de 2 digit posities waarbij de frequenties zo homogeen mogelijk verdeeld zijn: d4 en d7

Antwoord op 10d goed voor max 3 punten:
De hask key d4d7 levert de volgende rij hash waardes op:
17,59,68,55,72,36,40,49,72,38,17,91,60,24,15,05,17,12,82,56,33,04,07,59,24,97,45,60,44
Hiervan komt 17 3x voor en 59,24,60 en 72 2 maal; 6 botsingen dus.

Antwoord op 10e goed voor max 2 punten:
Aangezien de hash tabelgrootte 100 plaatsen kent [00..99] kan er bij botsingen tussen hashkeys van deze 29 studentID's altijd door uitwijken naar een 1^e vrije eronder dan wel erboven (cyclisch) een plek bepaald worden: 1^e 17 botsing -> 16, 2^e 17 botsing -> 14 (16,15 bezet); 59 -> 58; 24 -> 23; 60 -> 57 (59,58 bezet); 72 -> 71

11. Gegeven is de volgende adjacency matrix van de knopen a t/m f:

	a	b	c	d	e	f
a	0	0	0	1	0	1
b	0	0	1	0	0	1
c	1	1	0	0	0	0
d	0	1	0	0	0	0
e	0	0	1	1	0	0
f	1	0	0	0	1	0

a. Teken een gerichte graaf representatie van de gegeven adjacency matrix.
b. Voor welke 2 groepen knopen is dit een bipartite graaf?
c. geef de adjacency matrix van een bijbehorende ongerichte bipartite graaf (die aangeeft of er een relatie bestaat tussen 2 knopen uit de verschillende groepen).
(9 punten)

Antwoord op 11a goed voor max 2 punten:
Tekening van gerichte graaf.

Antwoord op 11b goed voor max 4 punten:
Bipartiet wil zeggen dat de groep knopen a t/m f te verdelen is in 2 subgroepen die onderling geen gerichte verbinding(en) kennen; de 2 groepen zijn (a,b,e) en (c,d,f);
Dit is goed in te zien als we van elke gerichte verbinding in de gegeven adjacency matrix een ongerichte verbinding maken:

	a	b	c	d	e	f	
a	0	0	1	1	0	1	a-> (c,d,f)
b	0	0	1	1	0	1	b->(c,d,f)
c	1	1	0	0	1	0	c->(a,b,e)
d	1	1	0	0	1	0	d->(a,b,e)
e	0	0	1	1	0	1	e->(c,d,f)
f	1	1	0	0	1	0	f-> a,b,e)

Antwoord op 11c goed voor max 3 punten:

	c	d	f
a	1	1	1
b	1	1	1
e	1	1	1

12. Gegeven de volgende 23 in lengte toenemende bitpatronen:
0,1,00,01,10,11,000,001,010,011,100,101,110,111,0000,0001,0010,0011,0100,0101,0110,0111,1000
a. waar moet een uniek decodeerbare verzameling codewoorden aan voldoen?
b. geef 2 mogelijke en verschillende verzamelingen Huffman codes voor de volgende 4 te coderen patronen aba(0.11), fgk (0.13), panama (0.26), plan(0.50) (getal tussen haakjes geeft fractie van voorkomen aan).
(8 punten)

Antwoord op 12a goed voor max 3 punten:
Van de gekozen codewoorden moet geen enkel begin overeenkomen met een al gekozen codewoord (van kortere lengte). B.v. 0,10,111 mag 0,10,011 niet

Antwoord op 12b goed voor max 5 punten:
Huffman codering neemt steeds de 2 kleinste fracties samen en hersorteert fracties tot er 2 over zijn;

plan 0.50	plan 0.50	plan 0.50
panama 0.26	panama 0.26	panama+fgk+aba 0.50
fgk 0.13	fgk+aba 0.24	
aba 0.11		

vervolgens wordt omgekeerd vanuit de 2 eindfracties de bitcode opgebouwd door een 1 en 0 bit als begincodebit toe te kennen aan de twee eindfracties en steeds bij een samengenomen stel de ene fractie een 1 en de ander een 0 toe te kennen; bij elke splitsing kan dit naar willekeur, zodat er veel verschillende codesets mogelijk zijn zoals:

(1)plan 0.50	(1)plan 0.50	(1)plan 0.50
(01)panama 0.26	(01)panama 0.26	(0)panama+fgk+aba 0.50
(001)fgk 0.13	(00)fgk+aba 0.24	
(000)aba 0.11		

maar ook:

(0)plan 0.50	(0)plan 0.50	(0)plan 0.50
(11)panama 0.26	(11)panama 0.26	(1)panama+fgk+aba 0.50
(101)fgk 0.13	(10)fgk+aba 0.24	
(100)aba 0.11		