

TDDC70/TDDC91 Datastrukturer och algoritmer Tentamen (TEN1) 2013-08-27, 14–19 (TER2)

Examinator: Tommy Färnqvist
Jour: Tommy Färnqvist (telefon 070 4547668).
Max poäng: 29 poäng (betyg 5 = 25p, 4 = 20p, 3 = 14p)
Hjälpmedel: INGA HJÄLPMEDEL TILLÅTNA!!!

VÄNLIGEN IAKTTAG FÖLJANDE

- Lösningar till olika problem skall placeras enkelsidigt på separata blad. Skriv inte två lösningar på samma papper.
- Sortera lösningarna innan de lämnas in.
- MOTIVERA DINA SVAR ORDENTLIGT: avsaknad av, eller otillräckliga, förklaringar resulterar i poängavdrag. Även felaktiga svar kan ge poäng om de är korrekt motiverade.
- Om ett problem medger flera olika lösningar, t.ex. algoritmer med olika tidskomplexitet, ger endast optimala lösningar maximalt antal poäng.
- SE TILL ATT DINA LÖSNINGAR/SVAR ÄR LÄSBARA.
- Lämna plats för kommentarer.

Lycka till!

1. Vilka av följande påståenden är sanna och vilka är falska? Svar utan motivering ger inga poäng. (3 p)

(a) Om $f(n) \in O(h(n))$ och $g(n) \in O(h(n))$, så gäller $f(n) \cdot g(n) \in O(h(n))$. (1)

(b) $n \log_2(n) \in \Omega(n)$ (1)

(c) Det finns en konstant k sådan att $3^{\log(n^2)} \in O(n^k)$ gäller. (1)

2. Hur ser splayträdet nedan ut efter: (4 p)

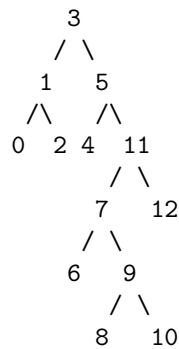
(a) `last()` (operationen som hittar det största elementet) (1)

(b) `insert(4.5)` (1)

(c) `find(10)` (1)

(d) `remove(9)` (1)

Utför varje deluppgift på originalträdet, inte trädet som resulterade från föregående operation.



3. Stackar och köer (6 p)

(a) Beskriv hur en initialt tom stack ser ut efter varje operation i följande sekvens av operationer: (2)

`push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop()`.

(b) Beskriv hur en initialt tom kö ser ut efter varje operation i följande sekvens av operationer: (2)

`enqueue(5), enqueue(3), dequeue(), enqueue(2), enqueue(8), dequeue(), dequeue(), enqueue(9), enqueue(1), dequeue(), enqueue(7), enqueue(6), dequeue(), dequeue(), enqueue(4), dequeue(), dequeue()`.

(c) Beskriv med pseudokod en algoritm med linjär tidskomplexitet för att vända på en kö Q . För att komma åt elementen i kön får du bara använda metoderna som finns i ADT `QUEUE`. I övrigt är det naturligtvis fritt att använda sig av extra strukturer av lämplig sort. (2)

4. Beskriv en effektiv implementation av ADT `DICTIONARY` för att lagra n element som har en associerad mängd av $r < n$ nycklar som kommer från en totalordning. Alltså, mängden nycklar är mindre än mängden element. Din implementation ska kunna utföra operationen `getAll` i $O(\log(r) + s)$ förväntad tid, där s är antalet element som returneras, operationen `entrySet()` i $O(n)$ tid och resterande operationer i ADT `DICTIONARY` i $O(\log(r))$ förväntad tid. (3 p)

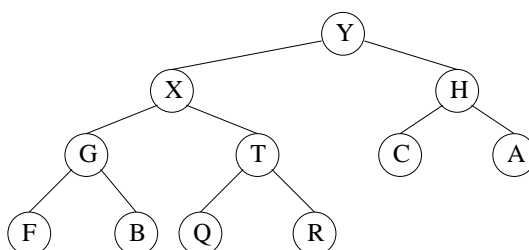
5. ADT MAP (3 p)

Antag att en applikation endast använder tre operationer, `insert()`, `find()`, and `remove()`.

- (a) Under vilka omständigheter skulle du använda ett splayträd istället för en hashtabell? (1)
- (b) Under vilka omständigheter skulle du använda ett (2, 4)-träd istället för ett splayträd? (1)
- (c) Under vilka omständigheter skulle du använda en osorterad array istället för ett (2, 4)-träd? (1)

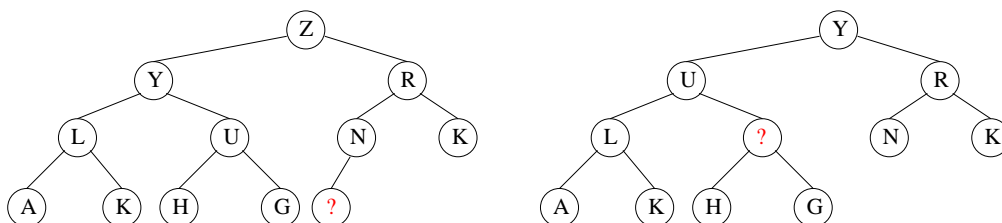
6. Nycklarna i den här uppgiften om binära heapar är stora bokstäver och vi använder bokstavsordning för att sortera dessa nycklar. (3 p)

- (a) Betrakta följande max-heap representerad som ett binärt träd. (1)



Visa hur en arrayrepresentation av heapen ser ut. (Du får själv välja om du vill använda en 0-indexerad eller 1-indexerad arrayrepresentation.)

- (b) Sätt in nyckeln P i den binära heapen ovan. Illustrera vilka operationer som utförs och hur slutresultatet ser ut. (1)
- (c) Ett anrop till `removeMax` genomförs på den binära heapen nedan till vänster. Resultatet av anropet illustreras av heapen nedan till höger. (1)



Vilka av nycklarna nedan skulle kunna vara nyckeln markerad med ett frågetecken i heaparna ovan?

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

7. Vi använder en array med indexen 0 till 6 för att implementera en öppet adresserad hashtabell av längd 7 med följande tabell som hashfunktion: (4 p)

nyckel	hashvärde
A	3
B	1
C	4
D	1
E	5
F	2
G	5

Antag att linjär sondering (*linear probing*) används för att hantera kollisioner.

(a) Visa hur arrayen ser ut efter att nycklarna har satts in i alfabetisk ordning: A, B, C, D, E, F, G. (2)

(b) Vilka av följande skulle kunna vara innehållet i arrayen efter att nycklarna har satts in i någon ordning? Varför/varför inte? (2)

I.

0	1	2	3	4	5	6
G	B	D	F	A	C	E

II.

0	1	2	3	4	5	6
B	G	D	F	A	C	E

III.

0	1	2	3	4	5	6
E	G	F	A	B	C	D

8. Konstruera en algoritm för att hitta det minsta antalet bågar som behöver tas bort från en given orientad graf så att den resulterande grafen är acyklisk. (3 p)