Några svar till
TDDC70/91 Datastrukturer och algoritmer

2013-01-12

Följande är lösningsskisser och svar till uppgifterna på tentan. Lösningarna som ges här ska bara ses som vägledning och är oftast inte tillräckliga som svar på tentan.

1. (a) **Falskt**. Ett motexempel: $t_1(n) = n^2, t_2(n) = n, f(n) = n^2$.

   (b) **Falskt**. Antag att påståendet är sant. Då finns konstanter $c > 0$ och n_0 sådana att för $n \geq n_0$ gäller: $4^{n-1} \leq c \cdot 3^n$. Dvs, $4^n \leq 4c \cdot 3^n$ och $(\frac{4}{3})^n \leq 4c$ för varje $n \geq n_0$, vilket inte kan vara sant eftersom c är en konstant. Alltså är påståendet falskt.

   (c) **Sant**. $2^{3\log_2(n)} = 8^{\log_2(n)} = n^{\log_2(8)} = n^3 \in \Omega(n^2)$.

2. (a) `j` kan bli så stort som $\mathtt{i}^2$, vilket kan bli så stort som $\mathtt{n}^2$. `k` kan bli så stort som `j`, vilket är $\mathtt{n}^2$. Exekveringstiden är därför proportionell mot $\mathtt{n} \cdot \mathtt{n}^2 \cdot \mathtt{n}^2$, vilket är $O(\mathtt{n}^5)$.

   (b) if-satsen exekveras som mest $\mathtt{n}^3$ gånger, enligt argumentet ovan, men är endast sann $O(\mathtt{n}^2)$ gånger (eftersom den är sann exakt `i` gånger för varje `i`). Alltså exekveras den innersta loopen bara $O(\mathtt{n}^2)$ gånger. Varje gång tar den $O(\mathtt{j}^2)$, eller $O(\mathtt{n}^2)$ tid, vilket ger totalt $O(\mathtt{n}^4)$ tid. Det fungerar inte alltid att multiplicera looparnas storlek.

3. (a) Kön kommer som mest att innehålla tre element samtidigt, så ringbuffern kan minst ha storlek tre.

   (b) 5, 3, 2, 8, 9.

   (c)
```
 B
[ | |1]
     F
```

   (d) Vi använder två stackar. En stack S används för att hålla reda på de `push` och `pop` som görs. Den andra stacken M håller reda på det minsta elementet. För att implementera `push` gör vi `push` på S. Om det nya elementet är mindre än översta element på M gör vi också `push` på M. För att implementera `pop` gör vi `pop` på S. Om elementet som `pop`:as är samma som det översta elementet på M `pop`:ar vi M också. `findMin` görs genom att undersöka översta elementet på M. All dessa operationer går uppenbarligen att utföra i tid $O(1)$.

   (e) Med dessa fyra operationer går det att sortera en indatasekvens med $O(n)$ operationer. Vi vet att sortering tar tid $\Omega(n \log n)$ vilket ger att minst en av operationerna måste använda $\Omega(\log n)$ tid.

4. (a)
```
              27                  18                22
        20          31        10        21        20
   13       23    30        5   16              13
 6   17                                       6   16
             AVL                 AVL               ej AVL
```

(b)

5. `1:a 2:e 4:d 5:f 6:c 7:b`

6. (a) Endast den slutliga arrayrepresentation redovisas här.

0	1	2	3	4	5	6	7	8	9	10	11	12
-	26	19	20	16	14	18	1	5	9	8	7	15

   (b) Sant.

7. (a) Endast slutresultatet redovisas här.

0	1	2	3	4	5	6	7	8	9	10
				`del`	4,e	26,b	5,d			

   (b) Endast slutresultatet redovisas här.

0	1	2	3	4	5	6	7	8	9	10
				`del`	4,e			26,b		5,d

8. Lösningsförslag: Välj en godtycklig nod v och färga den red. Färga alla dess grannar blue och kolla om någon konflikt uppstått. Om inte, färga alla noder som gränsar till de blue-färgade red och så vidare. Antingen lyckas man färga hela grafen eller så uppstår en konflikt någonstans. Det kan vara lite för mycket begärt att man ska prestera ett helt formellt korrekthetsbevis men en vettig diskussion kan man kräva.