

Några svar till TDDI16 Datastrukturer och algoritmer

2014-08-25

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. $\mathcal{O}(N^2)$. Läs in genomen går i linjär tid. For-looparna är helt enkelt insertion sort, vilken har ett kvadratisk värstfall (då break-satsen aldrig exekveras).
2. Returnerar true om Node är en pekare till roten av ett korrekt konstruerat BST.
3. - V J R E J G F F B F E A E B D
4. 1:a 2:e 3:d 4:f 5:c 6:b
5. (a) Eftersom M är mycket mindre än N är det absolut värsta fallet att vart och ett av de M elementen måste flyttas till början av originalarrayen. I det fallet blir körtiden $\mathcal{O}(MN)$.
(b) Att använda en array är en dålig idé om vårt mål är att nå optimal tidsanvändning eftersom vi, hur vi än försöker, fortfarande måste flytta de N gamla elementen. En bättre idé är att använda ett träd. Om vi använder ett balanserat binärt sökträd blir den maximala trädhöjden proportionell mot $\log N$, vilket medför att insättning av de M nya elementen exekverar i tid $\mathcal{O}(M \log N)$.
Mängden minne som går åt är $M + N$ vilket är optimalt inom en konstant faktor (eftersom vi måste lagra allt data).
6.
 1. D (trivialt att beräkna index via vinkeln)
 2. C (ingen naturlig ordningsrelation finns, bättre än oordnad array)
 3. B (operationerna har garanterad logaritmisk värstfallskomplexitet)