

Tentamen Datastrukturer och algoritmer, dt046g, dt064g

Martin Kjellqvist

2011-04-27

Instruktioner

Inga hjälpmedel

Maximalt en uppgift per papper. Du kan använda så många papper du önskar för varje uppgift. Skriv endast på en sida. Ange skrivkod på alla sidor. Om skrivkod inte finns tillgänglig skriver du ditt namn och personnummer på alla sidor. Skriv enkelt och tydligt för att undvika feltolkningar. Du får ej använda röd penna.

En uppgift med flera delar kan ge poäng utan att alla delar är lösta. Uppgifterna är inte ordnade efter svårighetsgrad. Läs igenom uppgifterna noggrant.

Preliminära gränser

A/B/C/D/E: 45/40/35/30/25 p

Max: 50 p

Lycka till!

/Martin

Uppgifter

1. Ordna följande uttryck efter deras asymptotiska tillväxttakt, långsammast växande först. Om uttrycket inte är ett polynom ange en övre gräns på polynomisk form i O notation.

$$0.2N^3$$

$$24N * \log(2^N)$$

$$N \log N$$

$$N^2$$

$$N\sqrt{\log N}$$

(4 p)

2. Betrakta ett binärt sökträd. Då man söker efter ett element, är då sökningen BFS-lik eller DFS-lik. Motivera / illustrera med exempel. (4 p)
3. Betrakta en prioritetskö implementerad som ett arraybaserat binärträd. Insättning av ett element görs på vanligt sätt på position N (0-baserat index). Beskriv en $O(\log N)$ algoritm som återupprättar heap-egenskapen för kön. Använd c++ kod eller pseudokod. Var noggrann med preconditions och postconditions för algoritmen. (5 p)
4. Som 3. men för borttagning av ett godtyckligt element. Notera att elementet inte nödvändigtvis är det med högst prioritet. (4 p)
5. Denial-of-service (DOS) attacker kan konstrueras mot en server som sorterar skickat data genom att man konstruerar sk. patologiska indata för sorteringsalgoritmen som används. Quicksort(höger pivot) degenererar till $O(N^2)$ i worst-case för indata som är enkla att konstruera och är därför olämplig att använda med tanke på DOS-attacker. En förbättring av Quicksort är som bekant Median-of-3. Är Median-of-3 ett gott skydd för att förhindra DOS-attacker pga patologiska indata? Om så inte är fallet vad vore ett bättre alternativ? Var utförlig i din diskussion. (7 p)
6. De flesta enkla (naiva) sorteringsalgoritmer uppvisar komplexiteten $O(N^2)$ vilket ofta gör dem olämpliga. Det finns emellertid några fall då dessa algoritmers beteende är att föredra. Beskriv två sådana fall. (4 p)

7. Divide-and-conquer är ofta en god metod då man behandlar data. Betrakta en osorterad array med N element. Beskriv en algoritm som finner det M -te minsta elementet med komplexiteten $O(N)$ $M < N$. Notera att utan $O(N)$ beteende ger ditt svar 0 poäng och konstanten till N kan vara stor men begränsad. (6 p)
8. Vilka blir delresultaten av att tillämpa en vanlig insertionsort på tecknen i strängen "examquestion"? Förklara delstegen. (6 p)
9. Vilken tidskomplexitet kommer följande kodavsnitt ha med avseende på parametern $size$? $Process(A, N)$ är $O(\log N)$. Motivera.

```
function Complex(array[], size):Integer{
    if ( size < 2 ) return size;
    result = Complex(array, size-2);
    result = result + Complex(array, size-1);
    return result + Process(array, size/2);
}
```

(6 p)

10. Vilka minimala förutsättningar måste gälla för att ett uppslag i en hash-tabell ska vara $O(1)$? Hur måste hashtabellen implementeras för att $O(1)$ beteendet garanteras efter en insättning. (4 p)