

Tentamen

dt046g, dt064g Datastrukturer och algoritmer

Martin Kjellqvist*

2012-04-23

Instructions

Läs igenom frågorna noggrant innan du börja besvara dem. Du har begränsat med tid, planera hur du ska besvara frågorna. Besvara endast det som efterfrågas. Skriv inte om saker som inte berörs av frågan.

Skriv svaren på erhållna svarspapper, inte på tentan. Varje ny fråga besvaras på ett nytt svarspapper. Skriv bara på en sida på svarspapperet.

Skriv tydligt. Om svaret är oläsligt får du 0 poäng - även om svaret är korrekt. Frågorna är *inte* ordnade efter svårighetsgrad.

Tid 5 timmar.

Hjälpmedel Inga.

Max poäng 46

Antal frågor 8

Preliminära gränser

$E \geq 40\%$, $D \geq 50\%$, $C \geq 60\%$, $B \geq 75\%$, $A \geq 90\%$.

Questions

- (4p) 1. Ordna följande uttryck efter deras tillväxttakt, långsammast växande först. Ange om uttrycket är ett polynom eller inte

- $N \cdot N \cdot N^2$
- $\frac{3 \cdot N}{N}$
- $N \log 2^N$
- \sqrt{N}

*martin.kjellqvist@miun.se

• 2^N

- (7p) 2. Det finns en algoritm beskriven i ett finurligt implementerat bibliotek. Tyvärr är dokumentationen utskriven på papper och någon har spillt kaffe på papperet så att det är oläsligt. Oläsliga tecken är angivna med *** eller inget alls. Algoritmen är dokumenterad som,

```

1 /**
2  * Algoitm addNeighboursAndVisit
3  * input:
4  * graph g – indexed graph to traverse
5  * int startingNodeIndex – root node for traversal
6  * nodeContainer s – storage container for visited node*****
7  * Use a stack container to implement the beh*****S
8  * search. *****
9  * Use a queue container to implement the behaviour o *****
10 * search. **
11 * Use a priorityQue*****ainer to implement the behaviour
12 * of the classic *ijk*****'*****h*.
13 * returns: **** *****
14 * A vector containing the*v*sited elements in the order
15 * presented by the nodeCo*tainer.
16 */

```

Vad står det i dokumentationen? Motivera så utförligt du kan.

- (3p) 3. Beskriv principen bakom en sk. xor-swap, dvs en implementation av swap som inte kräver en temporär lagringsyta. Ge exempel för några icke-triviala värden.
4. *Grafer*: terminologi och principer. Ge korta beskrivningar.
- (2p) (a) Vad innebär det att en *graf* är riktad?
- (2p) (b) Hur implementerar man en riktad *graf* i en adjacency matrix.
- (2p) (c) Ange någon *välkänd* algoritm som kan utnyttjas som metod för att konstruera ett träd utifrån ovan givna *graf*.
- (6p) 5. Vilka blir delresultaten av att tillämpa en vanlig quicksort på tecknen

examquestion

Förklara delstegen utförligt.

- (4p) 6. Beskriv två vanliga sätt att handskas med kollisioner i samband med insättning i en hash-tabell.
7. Sortering med prioritetsköer. Du har följande datastruktur.

```

1 class PriorityQueue
2 {
3 public:
4     PriorityQueue();
5     void Enqueue(int i); // log time insertion
6     int Dequeue(); // log time removal
7     int Head() const; // constant time

```

```
8 | bool IsEmpty() const;  
9 | };
```

- (4p) (a) Författa en algoritm som med hjälp av ovanstående struktur sorterar en array A .
 - (2p) (b) Vilken sorteringsalgoritm imiterar du?
 - (2p) (c) Vilken tidskomplexitet har den imiterade algoritmen? Motivera!
 - (2p) (d) Vilken tidskomplexitet har din algoritm? Motivera!
- (6p) 8. Implementera en komplett klass i c++ kod som representerar den prioritetskö som beskrivs i den tidigare uppgiften. Ta med samtliga beskrivna operationer.
Utifrån din implementation anger du komplexiteten hos de operationerna.

Lycka till,
Martin och Daniel.