

Tentamen

dt046g/dt064g Datastrukturer och algoritmer

Martin Kjellqvist*

2013-04-22

Instruktioner

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 50

Antal frågor 7

Preliminära gränser

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

Frågor

- (6p) 1. $O()$ -notationen kan vid en första anblick verka ge ett grovt och oprecist intryck. Vi har under kursens gång sett flera prov på dess användbarhet.
- Beskriv på vilka sätt $O()$ -notationen är användbar.
 - Vi har under kursen flera gånger konstaterat att om en algoritm tillhör $O(N)$ tillhör den också $O(N^2)$. Hur kan detta vara sant?
- (9p) 2. Följande problem och situationer har en tidskomplexitetskaraktär.
- Din uppgift är att identifiera komplexiteten i situationen. Du skall inte lösa problemen, de finns där för att illustrera en situation.
- I en tabell anger du komplexiteten med $O()$ notation för varje situation, beskriv i korthet ditt resonemang.
- Ordna tabellen så att den minst komplexa står först i tabellen, den mest komplexa sist.

*martin.kjellqvist@miun.se

A Du är på biblioteket och ska låna en bok: "C++ template metaprogramming : concepts, tools, and techniques from boost and beyond / David Abrahams, Aleksey Gurtovoy".

Biblioteket tillämpar SAB-systemet för klassificering. Du ser en bibliotekarie som ser ut att vilja svara på en fråga.

Hitta hyllan där din bok står.

B Du har en balansvåg med två skålar. Du har fått N stycken kulor. En av kulorna väger 1% mer än de övriga.

Hitta den tunga kulan.

C Du har blivit antagen till Fångarna på fortet".

Du blir ställd i en labyrint. För att klara dig ur labyrinten måste du passera N dörrar. Inför varje dörr måste du trycka på en vit eller svart knapp. Om du tryckt på fel knapp innan du går igenom dörren aktiveras en falllucka som bryskt transporterar dig till början av labyrinten.

Du har tre fall att betrakta:

C.1 De dörrar du passerar fortsätter att vara öppna. Räkna antalet knapptryckningar som är nödvändiga.

C.2 De dörrar du passerar stängs, men du kommer ihåg hur knapparna trycks ned.

C.3 Varje gång du åker tillbaka till början stängs dörrarna och knapparna konfigurerar om slumpmässigt

Ta dig ur labyrinten.

(9p) 3. För varje situation ovan, beskriv en algoritm eller datastruktur som presenterats under kursen som relaterar till situationen (eller åtminstone delar komplexiteten).

Namnge, beskriv och förklara algoritmen/datastrukturen. Ge en kort motivering hur du relaterar ditt val till situationen.

(8p) 4. Huffmanträd kan användas för att komprimera text.

Beskriv hur ett Huffmanträd ställs upp för följande text

1 Han kom som ett yrväder en aprilafton.

Beskriv i grova drag vilka egenskaper som krävs av ett filformat som implementerar denna typ av komprimering.

(2p) 5. (a) Varför tillåts inte vikter ≤ 0 i Dijkstras algoritm?

(2p) (b) Kan Dijkstras algoritm tillämpas på en digraph?

(6p) 6. Vilka blir delresultaten av att tillämpa en heapsort på tecknen

examquestion

Förklara delstegen. Din förklaring måste vara välstrukturerad och lättläst. varje upptänklig situation, fokusera på principerna och hur noderna arrangeras.

(8p) 7. Implementera en komplett klass i c++ kod som representerar en prioritetsskö. Prioritetsskön ska innehålla operationerna push(e) och pop():e, med vanlig semantik.

Beskriv vilken komplexitet de båda operationerna har i din kod.

Ange vilken komplexitet push och pop har optimalt. Om din lösning inte har denna komplexitet beskriver hur man går tillväga för att uppnå den.

Din enda restriktion är att du inte får använda klassen stack<T> ur std::.

Lycka till,
Martin