

Tentamen

dt046g/dt064g Datastrukturer och algoritmer

Martin Kjellqvist*

2016-01-15

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 8

Preliminära gränser

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

Frågor

1. Länkade listor är ofta beskrivna som en enkel och effektiv struktur för att genomföra insättningar. Vi har sett att vanliga arraystrukturer kan konstrueras så att de kan växa dynamiskt utan att uppföra sig patologiskt.
 - (2p) (a) Beskriv insättning av element i en dynamisk array så att insättning inte är $O(N^2)$.
 - (2p) (b) Beskriv insättning av element i en länkad lista så att insättningen är $O(1)$.
 - (4p) (c) Argumentera för eller emot om insättning av en miljon heltal går snabbare i en länkad lista eller i en dynamisk array.
- (3p) 2. Ordna följande uttryck efter asymptotisk tillväxttakt. Ange om uttrycket är ett polynom eller inte.
 - $N^2 \log 3^N$
 - \sqrt{N}

*martin.kjellqvist@miun.se

- $N^2 + 3N + \sqrt{3}$

- (9p) 3. 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.

A I den moderna filmatiseringen av "The day the earth stood still" ser vi en staty i Central park som skapar små flugor av inkringliggande material. Alla material verkar kunna användas i skapandet. Flugorna kommer sedan att konsumera allt i sin väg.

Flugorna verkar väga ungefär 10 gram och i den bildsekvens som visar hur flugorna fungerar ser man att varje fluga delar sig ungefär varannan sekund. Flugorna saktar inte ner sin fortplantning i senare generationer.

Hur långt är det kvar av filmen?

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.

C1 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. Dörrarna uppför sig lika vid nästa försök.

Ta dig ur labyrinten.

C2 Dörrarna konfigureras om efter varje försök.

- (4p) 4. För situationerna A och C 1, 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) 5. Huffmanträd kan användas för att komprimera text.

Beskriv hur ett Huffmanträd ställs upp för följande text

1 All your base are belong to us

Ange vilken kodning meddelandet får för ditt uppställda träd.

- (2p) 6. (a) Varför tillåts inte vikter ≤ 0 i Dijkstras algoritm?
 (2p) (b) Kan Dijkstras algoritm tillämpas på en digraph?
 (6p) 7. Vilka blir delresultaten av att tillämpa en heapsort på tecknen

examquestion

Förklara delstegen. Din förklaring måste vara välstrukturerad och lättläst.

- (8p) 8. Implementera en komplett klass i c++ kod som representerar en prioritetskö] Prioritetskön 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 `std::priority_queue<T>`.

Lycka till,
 Martin