

Tentamen i Datastrukturer och algoritmer, fk, DVA246

Akademien för innovation, design och teknik

2023-06-01

Skrivtid: 14.30 – 18.30

Hjälpmedel: Inga hjälpmedel

Lärare: För uppgift 2, 3, 4 & 5: Caroline kan nås på telefon kl 15.00-16.00
För uppgift 1, 6, 7, 8 & 9: Linus kan nås på telefon kl 15.00 – 16.00

Preliminära betygsgränser

Betyg 3: 50%

Betyg 4: 70%

Betyg 5: 90%

Max: 46p

Tentamen består av 9 uppgifter (observera att vissa av dessa består av deluppgifter)

Krav och allmänna råd:

- Läs igenom hela tentan för att veta hur du skall disponera din tid
- Frågorna står inte i svårighetsordning
- Skriv tydligt vilken uppgift du svarar på
- Ordna svaren i rätt ordning med svaret på uppgift 1 först och svaret på uppgift 8 sist
- Om du är osäker på vad som avses och gör antaganden skriv ut vilka dessa antaganden är.
- Det går att få poäng för partiella lösningar givet att de är meningsfulla

Uppgift 1: Korrekthet (6p)

Antag följande program som reverserar en array. Argumentera stringent för dess korrekthet, d.v.s., genom resonemang att finna en loop-invariant och visa att den gäller 1) från början, 2) under loop och 3) att loopen terminerar och då ger det förväntade resultatet. Notera att arrayerna är 0-indexerade.

```
int[] Reverse(int[] xs) {  
    var len = xs.Length;  
    var rs = new int[len];  
    for (var i = 0; i < len; i++) rs[len-1-i] = xs[i];  
    return rs;  
}
```

Uppgift 2: Heap (1+3p)

Deluppgift 2a:

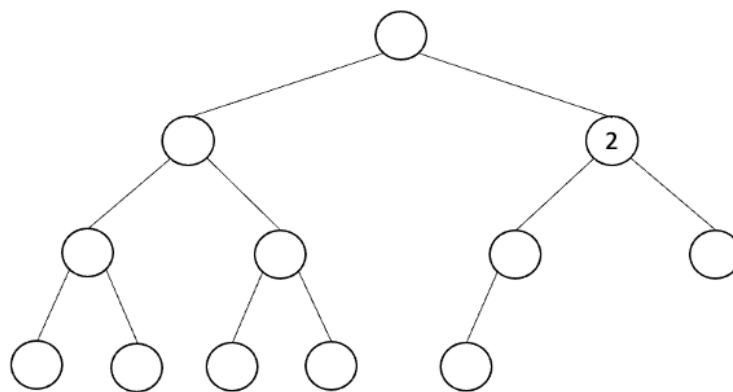
Vilken av följande arrayer representerar en binär max-heap?

Siffrorna representerar nycklarna i heapen. Nyckeln längst till vänster är index 0 i arrayen.

- a) 25, 12, 16, 13, 10, 8, 14
- b) 25, 14, 12, 13, 10, 8, 16
- c) 25, 14, 16, 13, 10, 8, 12
- d) Ingen av ovanstående

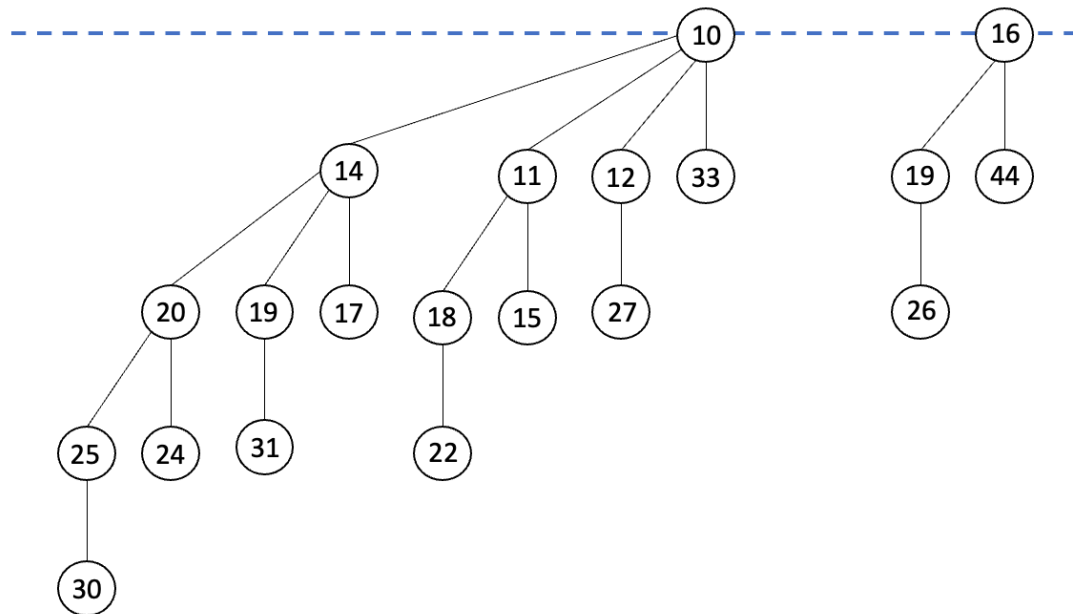
Deluppgift 2b:

Antag följande struktur på en binär min-heap. 2:an visar var det näst minsta nyckelvärde finns i strukturen. Rita av strukturen och skriv en 4:a i de noder som potentiellt kan innehålla det 4:e minsta nyckelvärde i heapen. Antag att heapen inte innehåller några dubletter.



Uppgift 3: Fibonacci-heap (7p)

Utgå från följande Fibonacci-heap. Siffran i noden representerar nyckeln. Den streckade linjen representerar root-listan. Fibonacci-heapen bygger på en min-heap:



Gör följande sekvens av operationer på Fibonacci-heapen:

`decreaseKey(30, 21)` //nyckel 30 minskas till 21

`decreaseKey(27, 9)`

`decreaseKey(31, 13)`

`decreaseKey(17, 5)`

`decreaseKey(24, 7)`

`insert(32)`

FIGUR A

`extractMin()`

FIGUR B

`decreaseKey(18, 17)`

`decreaseKey(25, 3)`

FIGUR C

Där det står "FIGUR A/B/C" i sekvensen ska du svara med en bild över hur Fibonacci-heapen ser ut just då, det ska vara tydligt vilken bild som avser vilket steg i sekvensen. Du får även visa de mellanliggande stegen om du vill. Det är dock endast stegen markerade med "FIGUR A/B/C" som kommer att bedömas.

När noder ska läggas på root-listan, ska de läggas längst till höger.

När hopslagning sker ska det göras från vänster till höger. Vid hopslagning görs en grad färdig innan hopslagning för nästa grad påbörjas.

Uppgift 4: Röd-svarta träd (2+2p)

Deluppgift 4a:

Det finns fem regler som alltid måste följas för röd-svarta träd.

En av dem är att varje nod antingen är röd eller svart.

Vilka är de andra fyra reglerna?

Deluppgift 4b:

Beskriv/förklara hur den längsta samt den kortaste pathen/vägen från root till NIL-löv i ett röd-svart träd förhåller sig till varandra. Ta stöd i det röd-svarta trädets regler.

Uppgift 5: Grafer (3+2p)

Deluppgift 5a:

Algoritmerna Kruskal's och Prim's tar båda fram ett minimum spanning träd.

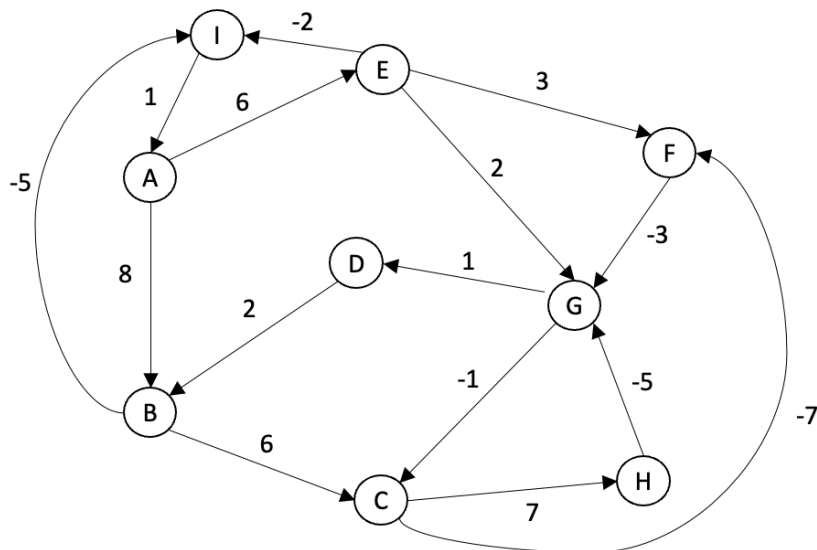
Beskriv vad ett minimum spanning träd är samt hur dessa båda algoritmer skiljer sig åt.

Deluppgift 5b:

Kan vi ta fram ett single source shortest path träd, med start i A, ur följande graf?

Om det är möjligt – visa hur trädet ser ut.

Om det inte är möjligt – förklara varför.



Uppgift 6: Giriga algoritmer (4p)

Kappsäcksproblemet är ett optimeringsproblem där det gäller att packa en kappsäck med ett antal föremål så att värdet på föremålen maximeras utan att överstiga kappsäckens maximala volym. Varje föremål har ett värde och en volym. I en av varianterna på kappsäcksproblemet gäller det att helt fylla kappsäcken samtidigt som man får dela föremålen i mindre delar. Det senare gör att en enkel girig algoritm är möjlig där man ordnar föremålen efter deras värde per volym och därefter fyller kappsäcken med så mycket av de mest värdefulla föremålen som får plats.

Bevisa att detta giriga val ger upphov till en maximal lösning genom ett enkelt motsägelsebevis.

Uppgift 7: Dynamisk programmering (4+4p)

Deluppgift 7a:

Vilka delar ingår i dynamisk programmering? Beskriv de egenskaper problemet och algoritmen har.

Deluppgift 7b:

Exemplifiera dynamisk programmering med valfri algoritm i kursen. Ge kod för algoritmen.

Uppgift 8: Rekursionsträd (4p)

För den rekursiva versionen av algoritmen du valt i uppgift 7b visa två rekursionsträd för lämplig indata. Det första trädet skall illustrera algoritmen utan dynamisk programmering och det andra trädet skall illustrera trädet med dynamisk programmering.

Uppgift 9: Substitutionsmetoden för rekurrensformler (4p)

Bevisa att $T(n) = 2T(\lfloor n/2 \rfloor) + n$ är $\Omega(n \lg n)$. Det räcker med att du bevisar för jämna potenser av 2.