

# Tentamen i Datastrukturer och algoritmer, fk, DVA246

Akademien för innovation, design och teknik

2023-01-10

**Skrivtid:** 14.30 – 18.30

**Hjälpmedel:** Inga hjälpmedel

**Lärare:** Caroline Uppsäll, kan nås på telefon: 021-101456 (kl 15.00 – 16.00)

## **Preliminära betygsgränser**

Betyg 3: 50%

Betyg 4: 70%

Betyg 5: 90%

Max: 49p

Tentamen består av 9 frågor.

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

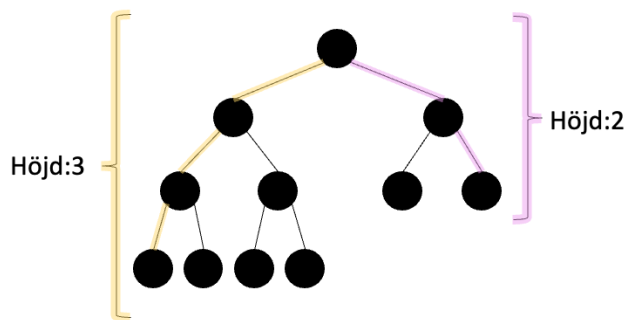
### Uppgift 1: Korrekthet (6p)

Förklara stegen i ett bevis av en loops korrekthet.

### Uppgift 2: Heap (2+2p)

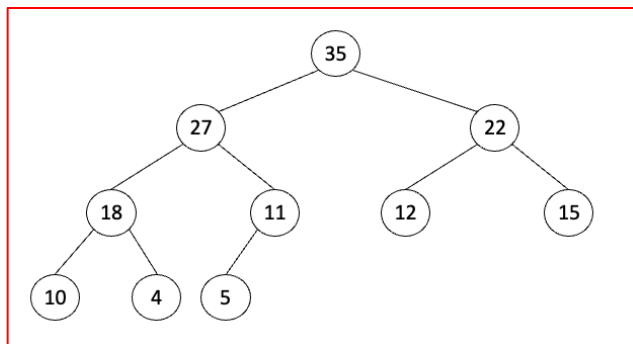
#### Deluppgift 1

Hur många element finns det som minst respektive som mest i en min-heap med höjd  $h$ ? Höjden  $h$  representeras av antalet båggar i den längsta pathen från rooten, se bild för exempel:

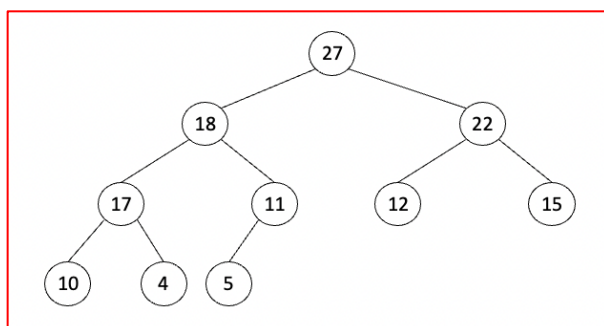


#### Deluppgift 2

Visa hur max-heapen ser ut efter att följande nycklar sats in (i angiven ordning): 15, 10, 27, 18, 5, 12, 22, 35, 4, 11



Visa sen hur heapen ser ut efter att nyckel 35 ändras till 17.



### Uppgift 3: Fibonacci-heap (4+3p)

#### Deluppgift 1

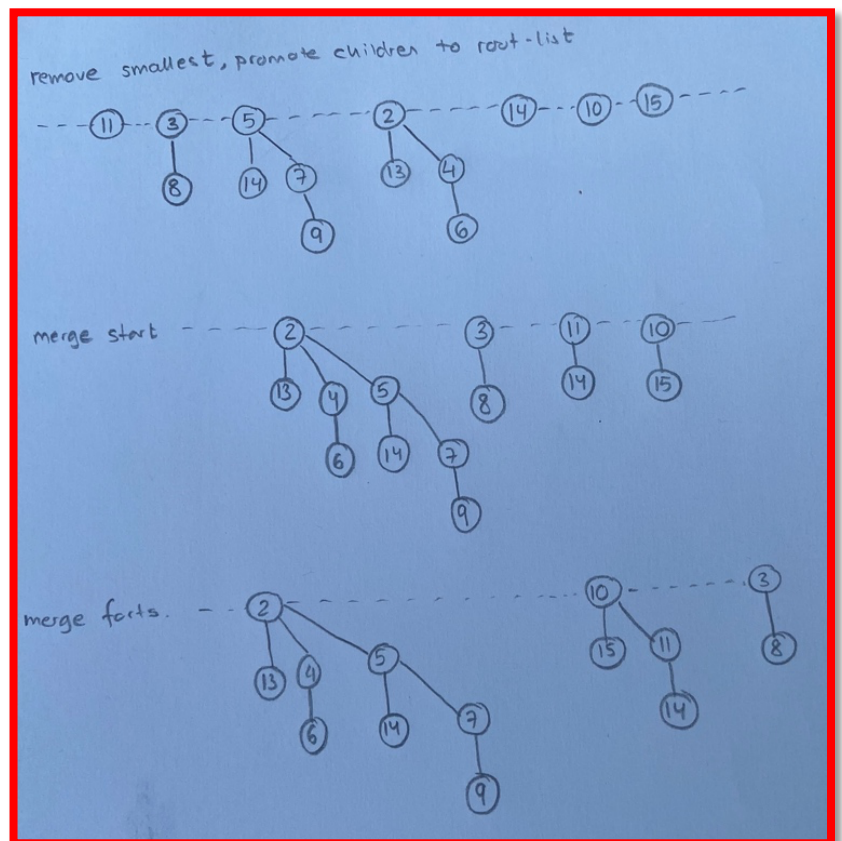
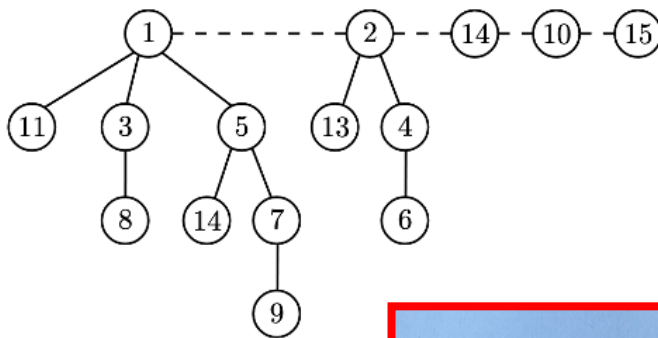
Antag att vi förenklar fibonacci-heapen på så sätt att markeringen (den en nod får när den förlorar ett barn) tas bort. Då kommer inte längre föräldrar som förlorat två barn flyttas till root-listan. Är den amorterade kostnaden för

- decreaseKey-operationen fortfarande  $O(1)$ ?
- extractMin-operationen fortfarande  $O(\log n)$ ?

Förklara/motivera dina svar.

#### Deluppgift 2

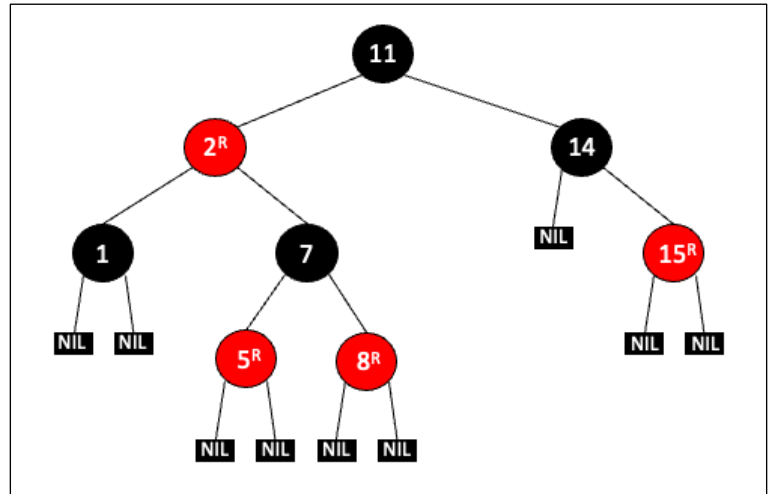
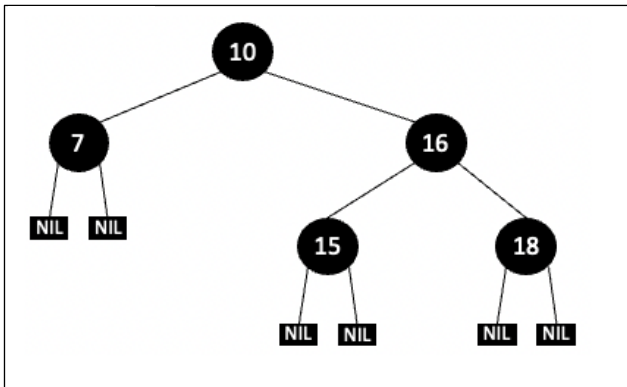
Antag fibonacci-heapen nedan, visa hur heapen ser ut efter att du har utfört en extractMin-operation. Förklara steg för steg hur du gjort. Om du behöver så slå samman träd ska det göras från höger till vänster.



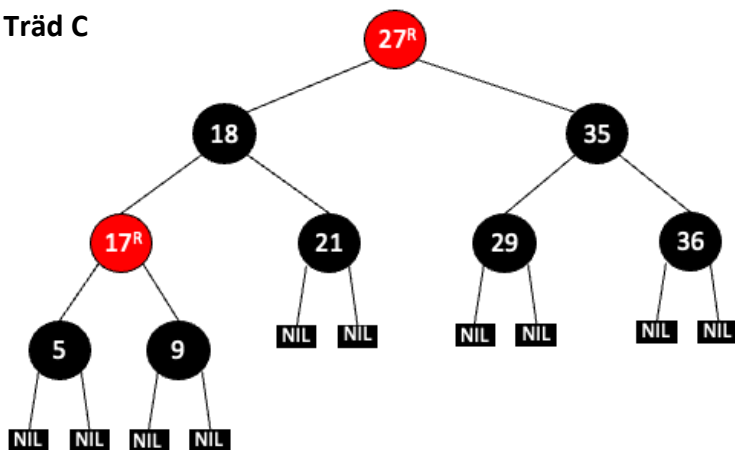
#### Uppgift 4: Röd-svarta träd (4p)

Är följande träd (Träd A, Träd B, Träd C och Träd D) rödsvarta träd? Motivera/förklara dina svar.

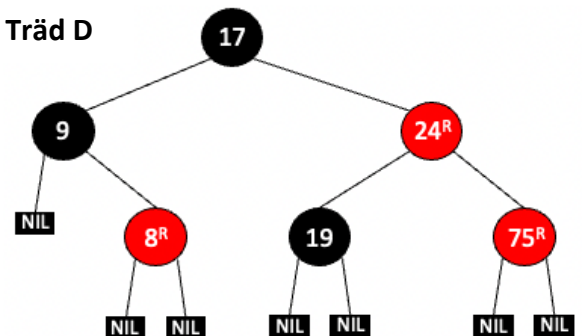
Motiveringen/förklaringen för ett träd ska vara komplett för att ge poäng.



Träd C



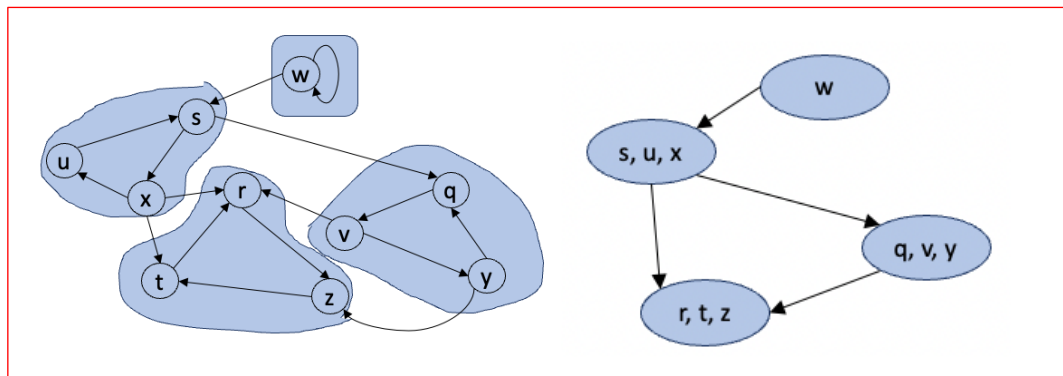
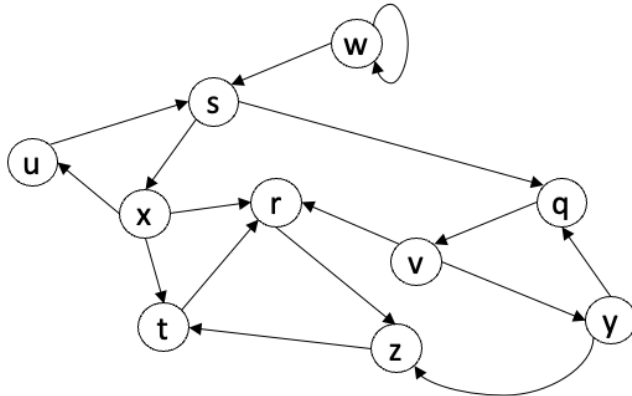
Träd D



## Uppgift 5: Grafer (3+3p)

### Deluppgift 1

Vilka strongly connected components består följande graf av, och hur relaterar komponenterna till varandra? Beskriv eller rita.



### Deluppgift 2

Antag att vi har  $n$  st objekt samt olika avstånd mellan objekten, vi kan då representera objekten som noder i en graf och avstånden som viktade bågar i grafen. Ponera nu att vi vill gruppera dessa objekt på ett sådant sätt att objekten inom en grupp har små avstånd men avstånden mellan grupperna är större. Vi kan kalla detta för klustering ("clustering").

Ett sätt att beräkna klustering på i en oriktad och viktad graf är genom att använda Kruskal's algoritm men stoppa algoritmen innan det har tagit fram ett fullt minimum spanning träd. Om vi stoppar algoritmen efter att  $k$  st bågar har lagts till i trädet, hur många komponenter (eller kluster) kommer vi då ha? Förklara hur du kom fram till ditt svar (använda max 6 meningar)

Tips: Från start är varje nod i grafen ett eget kluster.

#### Uppgift 6: Giriga algoritmer (6p)

Motsvarar en given frekvensindelning alltid ett unikt Huffmanträd? Om ja, argumentera för varför. Om nej, ge ett exempel på en frekvensindelning och ge två träd som motsvarar frekvensindelningen.

Svar: En frekvensindelning leder inte nödvändigtvis till unika Huffmanträd. Om fler än två kvarvarande delträd, säg a, b, och c, har samma frekvens kan vi välja att slå ihop a och b, b och c eller a och c. Detta steg ger sålunda upphov till tre möjliga träd.

#### Uppgift 7: Dynamisk programmering (4+4p)

Dynamisk programmering baserar på två typer av memoization. Vilka är det och vad är den grundläggande skillnaden mellan dem? Get en implementation av Fibonacci i båda varianter.

#### Uppgift 8: Rekursionsträd (4p)

Använd ett rekursionsträd för att uppskatta en bra asymptotisk övre begräsning för  $T(n) = T(n/3) + T(2n/3) + c(n)$

#### Uppgift 9: Substitutionsmetoden för rekurrensformler (4p)

Använd substitutionsmetoden för att visa att  $T(n) = T(n/2) + n^2$  är  $O(n^2)$