

DD1352 Algoritmer, datastrukturer och komplexitet, hösten 2015

Mästarprov 1: Algoritmer

Mästarprovet ska lösas **individuellt** och redovisas både skriftligt och muntligt. *Inget samarbete är tillåtet, se vidare hederskodexen.* Du ska alltså inte diskutera lösningar med någon annan fram till dess att alla muntliga redovisningar är avklarade.

Skriftliga lösningar ska lämnas senast **tisdag 13 oktober klockan 12.15** på föreläsningen eller senast klockan 12.00 samma dag i kursens inlämningslåda inne på receptionen, Osquars backe 2, plan 4. Det är viktigt att du lämnar in i tid!

Skriv ditt namn och personnummer överst på framsidan av lösningarna. Se till att spara en kopia av dina lösningar så att du kan läsa på inför den **muntliga redovisningen** som kommer att ske 19–26 oktober för någon av lärarna. Boka tid för en femton minuters muntlig redovisning på kurswebbsidan senast 13 oktober klockan 12. Bokningslistorna läggs upp senast 9 oktober. Om du inte hinner göra uppgiften så avbokar du enkelt din bokning.

Det är viktigt att du förbereder dig inför den muntliga redovisningen. För att en uppgift ska godkännas ska du kunna förklara och motivera algoritmen muntligt och reda ut eventuella oklarheter.

Läs uppgifterna mycket noga så att du inte råkar basera dina lösningar på en missuppfattning. Fråga en lärare på kursen om något är oklart.

Mästarprovet är ett obligatoriskt moment i kursen. Det består av tre uppgifter som motsvarar betygsriterierna för E, C respektive A. För godkänt (betyg E) krävs helt rätt på en av uppgifterna. Helt rätt på två av uppgifterna ger betyg C och alla rätt ger betyg A. Ett mindre fel på en uppgift sänker betyget ett steg. Läs mer om betyg på kurswebben.

För att se exempel på hur utförliga lösningarna bör vara kan du titta på lösningar till *tidigare mästarprov* på kurswebben. Där finns också tips om hur man skriver pseudokod, som du bör läsa innan du formulerar dina algoritmer. Några dagar före inlämningen kommer noggrannare kriterier för hur bedömningen går till att publiceras på kurswebben.

1. Hantverkare

Betygskriterium: utveckla algoritmer med datastrukturer för enkla problem givet en konstruktionsmetod.

Ditt kök behöver renoveras och n stycken hantverkare har just kommit till din lägenhet. Deras respektive uppdrag tar t_1, t_2, \dots, t_n tid att utföra.

De dåliga nyheterna är att inte mer än en person kan jobba samtidigt och att du måste betala både för tiden som de jobbar och tiden som de väntar på att kunna utföra sitt jobb.

De goda nyheterna är att du får bestämma i vilken ordning jobben ska utföras och att alla debiterar det enastående låga priset 100 kronor per timme.

Designa en girig algoritm som beräknar den minimala kostnaden för renoveringen. Beskriv algoritmen med pseudokod, förklara varför den fungerar och analysera dess tidskomplexitet.

2. Höravstånd

Betygskriterium: utveckla algoritmer med datastrukturer för icke-triviala problem.

Det står n personer på heltalspunkter på tallinjen. Två personer är inom hörhåll om deras avstånd är strikt mindre än k . Vi vill flytta på personerna så att alla par av personer kan kommunicera med varandra (möjligtvis via en eller flera andra personer) och minimera summan av sträckorna personerna flyttar på sig. Du kan anta att personerna inledningsvis finns i heltalskoordinater mellan 0 och $2kn$. Flera personer kan stå på samma koordinat.

Konstruera en effektiv algoritm som tar indata n, k och $P[1..n]$ (där $P[i]$ talar om var person i står från början; $P[1] \leq P[2] \leq \dots \leq P[n]$) och bestämmer minimala summan av sträckor personerna behöver flytta för att höra varandra.

Beskriv algoritmen med pseudokod och analysera dess tidskomplexitet uttryckt i n och k . Motivera också att algoritmen är korrekt.

3. Mängder av anagram!

Betygskriterium: utveckla algoritmer med datastrukturer för svårare problem med den metod som passar bäst.

Två ord är anagram av varandra om det går att kasta om bokstäverna i det ena så att det andra ordet bildas, som till exempel *logaritm* och *algoritm*.

Indata i denna uppgift är en ordlista med m stycken svenska ord (skrivna med små bokstäver, a-ö), där det längsta ordet har längd n . Anta att m och n är oberoende av varandra (så till exempel skulle det längsta ordet kunna vara längre än antalet ord, eller också är längsta ordet kortare än antalet ord).

Konstruera en algoritm som matar ut alla mängder av anagram i ordlistan, en mängd per rad. Alla ord som är anagram av varandra ska stå på samma rad, i bokstavsordning. Ordningen mellan raderna spelar däremot ingen roll. Ett ord som inte har något anagram ska stå ensamt på raden.

Om indata består av orden *algoritm*, *komplexitet*, *rotad*, *dator*, *logaritm*, *ordat* skulle utdata kunna se ut så här:

```
algoritm logaritm
dator ordat rotad
komplexitet
```

Beskriv algoritmen med pseudokod och analysera tidskomplexiteten. För godkänt krävs en algoritm som har optimal tidskomplexitet (uttryckt i n och m). Visa därför att övre och undre gräns sammanfaller. En i övrigt helt korrekt lösning med en algoritm som är en log-faktor ifrån undre gränsen ger sänkning med ett betygssteg.

Du ska motivera att din algoritm är korrekt.