

2D1320, TENTAMEN I TILLÄMPAD DATALOGI
Måndagen den 23 oktober 2006 kl 8–13

Maxpoäng = 100. Fler än 49 poäng ger godkänt. Skriv upp antal bonuspoäng från labbar respektive hemtal på tentaomslaget. Hjälpmedel : en algoritmbok och det handskrivna formelbladet. Lämna in formelbladet tillsammans med tentan. Tentorna återlämnas på föreläsningen 31/10 klockan 8-10 i sal K1. Sedan finns de på studentexpeditionen.

1. *Sorterade CD:s*. Arun har 1 miljon CD:s på ett plan i sin väldigt breda bokhylla. Varje gång han ska lyssna på en skiva tar han ut den och flyttar alla skivor som står till höger om den ett steg åt vänster. När han lyssnat klart ställer han skivan längst till höger. På så vis kommer de skivor han lyssnar oftast på att finnas längst till höger i bokyllan!
 - (4p) a. Det ovanstående liknar en stack. Vad är det som inte stämmer? Nämn två saker.
 - (2p) b. När Aruns kompis Magnus är på besök sorterar han om skivorna efter artist/band och utgivningsår. Vilken sorteringsmetod ska han använda för att det ska gå så fort som möjligt?
 - (6p) c. När hyllan är sorterad är det lätt att hitta en CD. Vilken/vilka av följande metoder *kan* användas? *linjärsökning, breddenförstsökning, heapsort, binärsökning, hashning, binärträdssökning, binärträdsgenomgång, Rabin-Karp* Vilken metod tar kortast tid? Vilken är dess tidskomplexitet? Om det fanns 32768 CD:s i hyllan, hur många måste man titta på i värsta fallet om man använde den metoden? Förklara.
 - (8p) d. Magnus är ändå inte riktigt nöjd. Han vill göra en grov indelning i följande tre genrer: klassiskt, jazz och pop. Vilken sorteringsmetod ska Magnus använda nu för att det ska gå så fort som möjligt? Skriv ned algoritmen. Vilken är dess komplexitet? Motivera med hjälp av en kort tidsanalys.
 - (4p) e. En dag inventerar Arun sin CD-samling. Han vill veta hur många låtar som finns totalt i den. Betrakta hyllan som en stack och skriv en rekursiv tanke för att ta reda på det. Antalet låtar på varje CD finns enkelt åtkomligt på baksidan.
 - (5p) f. När man lagrar informationen om en CD i ett program är det bäst att använda en abstrakt datatyp (ADT). Varför? Motivera utförligt. Definera ett lämpligt gränssnitt för denna ADT! Ange åtminstone tre lämpliga funktioner och vad de är tänkta att göra.
- (6p) 2. *“lay all your love on me”* sjunger ABBA. Konstruera en huffmankod för tecknen (bortse från mellanslagen) i denna önskan och skriv sedan upp den i kodad form. Rita huffmanträdet!
- (4p) 3. *Greatest Hits*. Skriv ett reguljärt uttryck för textsökning som träffar bla: *Greatest_hits, Greatest_Hits, greatest_Hits, greatest_hits*. Det ska alltså känna igen stort och litet h och g i inledningen till de två orden. Dessutom vill man träffa alla varianter med eller utan s:et på slutet.

4. *Bästa spellistan.* Istället för att slumpa ordningen på låtarna i en spellista skulle man kunna spela de bästa låtarna först. Om alla låtar är betygsatta på en skala från ett till fem (bäst) kan man använda en prioritetskö. Från den tas då gång på gång den bästa av de kvarvarande låtarna ut. Detta upprepas tills den är tom.
- (6p) a. Vilken datastruktur brukar användas för att implementera en prioritetskö? Stoppa in följande tal i denna ordning i en sådan datastruktur: 1, 4, 2, 5, 4, 3. Rita upp hur den ser ut efter varje insättning. Rita på båda sätten!
- (2p) b. Om man vill att låtarna i spellistan ska kunna återkomma (stoppas in i spellistan igen) är det inte en bra idé att använda en prioritetskö. Varför?
- (4p) c. Man kan förstås prioritera låtarna på många olika sätt. Ge förslag på en funktion som beräknar en låts prioritet med hjälp låtens betyg och antalet gånger den spelats. Här följer ett funktionsskelett att utgå från.

Python: <pre>def f(betyg, antal): ...</pre>	Java: <pre>public int f(int betyg, int antal){ ... }</pre>
--	---

Skriv en så enkel funktion som möjligt och motivera den kort.

5. *Influenser.* Alla musikartister/band har förebilder som påverkat dem. I en datastruktur har man sparat vilka som påverkat vilka. Anropet

```
influenser('Rolling Stones')
```

returnerar en kö som innehåller alla som påverkat *Rolling Stones*, bland annat den gamla blueslegenden *Muddy Waters*. Artisterna och banden lagras som strängar, tex: "Jimi Hendrix".

- (10p) a. Varje sann diskofil vill förstås också enkelt kunna få reda på det omvända: vilka andra har en artist påverkat? Man vill alltså att anropet
- ```
influerat('Muddy Waters')
```
- ska returnera en kö som bland annat innehåller *Rolling Stones* och *Jimi Hendrix*.  
Din uppgift är att bygga upp en ny datastruktur som ska användas för att `influerat`-funktionen ska returnera så snabbt som möjligt. Beskriv hur du går till väga. Utnyttja `influenser`-funktionen. Vilken/vilka datastrukturer använder du?  
Hur fungerar det sedan när någon använder `influerat`-funktionen?
- (16p) b. En bluesintresserad tildastudent undrar nu: vilken är den längsta influenskedjan utan "cirkel-influenser" från *Muddy Waters*? Hon vill naturligtvis gärna ha kedjan utskrivna.  
Beskriv en effektiv algoritm för att lösa uppgiften. Vilka datastrukturer behövs? Utgå från att `influerat`-funktionen fungerar som den beskrivs i a-uppgiften och använd den.

6. *Poptextssyntax*. Populärmusiktexter skiljer sig i många avseenden från vanlig text.

- (8p) a. Bland annat finns många *utropsord* som får dyka upp var som helst i en poptext, tex *yeah, oh, huh, well, boy*. Dessutom kan de upprepas hur många gånger som helst där de stoppas in. I princip tillåter man alltså en poptext med ett oändligt antal utropsord mellan två på varandra följande “vanliga” ord. Därför kan tex den normala meningen “I still got the blues” i en poptext bli:

Well I still yeah yeah oh got the oh blues oh boy

Skriv en syntax som beskriver poptexter! Du behöver inte bry dig om radbyten och du kan använda följande icke-slutsymboler utan motivering:

<yeah> - för ett utropsord

<ord> - för vilken annan följd av ord som helst

- (8p) b. Dessutom innehåller populärmusiktexter många upprepningar. Ett sätt att komprimera sådana texter är därför en slags följdängdskodning. Varje gång något upprepas skriver man det upprepade inom hakparenteser med en siffra innan som talar om hur många gånger det ska upprepas. Slutet av en känd låt av Beatles blir då:

... and you know you should be glad oh

2[She loves you yeah yeah yeah]

3[With a love like that you know you should be glad]

2[yeah yeah yeah] yeah!

Bygg vidare på syntaxen från a-uppgiften så att den nu gäller komprimerade poptexter. Svara med hela syntaxen. Du behöver fortfarande inte bry dig om radbyten.

*Man måste ha rätt på a-uppgiften för att kunna få några poäng på denna deluppgift.*

7. *Teori*. Några korta teorifrågor.

- (2p) a. Vilken tidskomplexitet har damerna-först-sortering? Motivera mycket kort.
- (2p) b. Rangordna följande fem funktioner efter hur snabbt de växer då  $n$  är stort:  $3n$ ,  $5$ ,  $7^{n^2}$ ,  $\sqrt{n}$ ,  $0.004n^2$ . Ange i vilken ordning du skriver dem!

Svara bara ja eller nej på följande tre frågor.

- (1p) c. Kan ett problemträd vara binärt?
- (1p) d. Kan man vända på en kö med hjälp av en stack?
- (1p) e. Är sökning i ett obalanserat binärt sökträd långsammare än binärsökning i en sorterad vektor i värsta fallet?