

DD1320/DD1321/DD1325  
TENTAMEN I TILLÄMPAD DATALOGI  
Torsdag 7 juni 2018 kl 14–18



Hjälpmedel: Valfri algoritmbok och ett egenskrivet formelblad. För betyg E krävs att alla E-uppgifter är godkända (upp till två E-uppgifter kan kompletteras). För betyg D krävs (utöver E-kraven) D på bägge C-uppgifterna, för C krävs minst ett C och ett D (en C-uppgift kan kompletteras till D). För betyg B respektive A krävs (utöver C-kraven) betyg B respektive A på A-uppgiften (A-uppgiften kan inte kompletteras). Lycka till!

**E 1.** *KTHB undrar hur bloomfilter kan användas.*

10 min

Biblioteket har en mycket stor fil med bokdata som t.ex. ISBN-nummer, titel, författare etc. Man vill kunna söka snabbt efter data och har två frågor till dig.

- Kan man använda bloomfilter för att snabbt ta reda på om biblioteket har en specifik titel? **Ja, man kan söka efter om något finns i ett bloomfilter. Det finns en mycket liten chans att man får en false positive, d.v.s. att en bok finns fast den inte gör det.**
- Kan man använda bloomfilter för att givet ett ISBN-nummer snabbt ta fram författaren? **Nej, man kan bara ta reda på om något finns. Annan data finns inte med.**

**E 2.** *KTH:s Incident Response Team vill veta vad du kan om kryptering.*

10 min

En krypteringmetod som användes innan datorer blev vanliga är *bokchiffer*.

- Beskriv och motivera en signifikant fördel med att använda bokchiffer som kryptering. **Det är mycket svårt att knäcka bokchiffer**
- Beskriv och motivera en signifikant nackdel med att använda bokchiffer som kryptering. **Att distribuera nyckeln (boken). Det kan vara tidsödande att avkoda manuellt**

**E 3.** *Bioteknik har komprimerat DNA.*

10 min

För att komprimera en fil med DNA-data har vi använt oss av Huffmankodning enligt följande tabell:

Tecken	Frekvens	Huffmankod
A (Adenin)	25%	11
G (Guanin)	25%	10
C (Cytosin)	25%	01
T (Tymin)	25%	00

Komprimeringsgraden blev sämre än väntat. Förklara varför!

**Huffmankodning är en statistisk metod. Om det är samma fördelning på bokstäverna så vinner man inte lika mycket.**

**E 4.** *KTH:s IT-support använder reguljära uttryck.*

Man har ett reguljärt uttryck

`[a-z]*@kth[.]se`

10 min

Vilka av nedanstående fyra strängar matchas? Motivera dina svar.

- a) alexander.baltatzis@kth.se **Nej, punkten matchar inte**
- b) alba@kth.se **Ja**
- c) STENE@kth.se **Nej, stora bokstäver matchar inte**
- d) @kth.se **Ja, \* matchar 0 eller fler**

**E 5.** *Avdelningen för Matematik har ett problem.*

10 min

Nedanstående rekursiva funktion `upphojt_till` fungerar inte. Vad är det som är problematiskt? Motivera ditt svar.

```
def upphojt_till(x, y):  
    return x * upphojt_till(x, y - 1)
```

```
def main():  
    z = upphojt_till(5, 3) # exempelanrop
```

**Den rekursiva funktionen har inget avbrottsvillkor och kommer fortsätta tills programmet kraschar**

```
def upphojt_till(x, y):  
    if y < 1:  
        return 1  
    return x * upphojt_till(x, y - 1)
```

```
def main():  
    z = upphojt_till(5, 3) # exempelanrop  
    print("5^3=", z)
```

**E 6.** *JML-gruppen sorterar.*

15 min

Vi vill sortera nedanstående element med sorteringsmetoden *damerna först*. Gröna element (G) ska stå till höger och blåa (B) element till vänster i den sorterade vektorn.

B G B G G B B G B G B G B B

a) Visa hur vektorn ser ut i varje steg av sorteringen.

```
['B', 'G', 'B', 'G', 'G', 'B', 'B', 'G', 'B', 'G', 'B', 'G', 'B', 'B']  
 * *  
['G', 'G', 'B', 'G', 'G', 'B', 'B', 'G', 'B', 'G', 'B', 'B', 'B', 'B']  
 * *  
['G', 'G', 'G', 'G', 'G', 'B', 'B', 'G', 'B', 'B', 'B', 'B', 'B', 'B']  
 * *  
['G', 'G', 'G', 'G', 'G', 'G', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B']  
 * *
```

b) Vilken tidskomplexitet har algoritmen? Motivera ditt svar.  
O(N)

**C 7.** *Industriell ekonomi funderar över effektiviteten.*

30 min

Alla Svenska företag (strax över en miljon, dvs ungefär  $2^{20}$ ) ska lagras i ett register. Följande kan vi behöva göra:

- **INSÄTTNING:** Varje dag tillkommer 10 nya företag som genast ska läggas till (vi behöver inte kontrollera om företaget redan finns med).
- **SÖKNING:** En gång varje månad frågas det om ett visst företag finns med i registret.
- **SÖKNING + BORTTAGNING:** Varje månad avregistreras 50 företag, dessa ska tas bort i slutet av månaden.

Jämför *antalet operationer* för följande tillvägagångssätt. För betyg C krävs att du gör överslagsberäkningar och jämför konkreta värden.

- a) Företagen lagras i en *vektor* (Pythonlista). Varje dag läggs företag till sist i vektorn (utan att sortera). I slutet av månaden måste vi söka i den osorterade vektorn efter de företag som avregistrerats och ska tas bort.
- b) Företagen lagras i en *länkad lista*. Varje dag läggs företag till sist i listan (utan att sortera). I slutet av månaden måste vi söka i den osorterade listan efter de företag som avregistrerats och ska tas bort.
- c) Företagen lagras i en *sorterad vektor*. Anta att vektorn är sorterad från början. Varje nytt företag sorteras in i vektorn så att man kan söka med binärsökning när företag ska tas bort.

Visa alla beräkningar och motivera dina svar.

**C 8.** *Arkitektur bygger slott.*

30 min

Ett magiskt sagoslott har många rum som är förbundna med varandra. En del rum angränsar till väldigt många rum och andra rum gränsar bara till ett enda. Ibland så ändras planlösningen magiskt. Det som händer då är att *antingen* så försvinner en förbindelse mellan två rum *eller* så tillkommer en ny förbindelse mellan två rum.

En tildastudent får uppdraget att lagra hur rummen är förbundna med varandra i en graf. Man ska kunna

- se om två rum är förbundna med varandra,
- se vilka rum som är förbundna med ett givet rum,
- uppdatera grafen när magi inträffar (rumsförbindelse försvinner/tillkommer).

- a) Rita och beskriv kortfattat hur grafen kan representeras som en grannlista. Rum kan vara siffror. T.ex

1 -> 2, 3, 5  
2 -> 1, 3  
3 -> 4, 2, 3  
4 -> 3  
5 -> 1

- b) Rita och beskriv kortfattat hur grafen kan representeras som en grannmatris. Rum kan vara siffror som fungerar som index, t.ex.

	1	2	3	4	5
1		x	x		x
2			x		
3				x	
4					
5					

- c) Jämför grannlista och grannmatris med avseende på prestanda och minnesåtgång för de tre punkterna ovan.

Minne - Om det är få vägar så är grannlistan billigare. Matrisen är lika stor alldeles oavsett.

Prestanda - Magin är mycket billig  $O(1)$  i matrisen, oavsett borttagning eller tillkomst så ändras en ruta i matrisen. Att lägga till en förbindelse i rumslistan är  $O(1)$  men att ta bort kostar om man ska söka  $O(N)$  upp förbindelsen, själva borttagningen är  $O(1)$ .

**A 9.** *Bioteknik återkommer med nya frågor.*

**Betyg A.**

*45 min*

En DNA-sekvens består av en följd kvävebaser, t ex

GTCCAACGGATCGAGATCTGAGCGTTCAAAGATCCGTAGATCTCAACCTC

Vissa sekvenser av tre eller fyra tecken är vanligare än andra, t ex "GATC" i sekvensen ovan:

GTCCAACG GATCGA GATCTGAGCGTTCAAAG GATCCGTA GATCTCAACCTC

- a) Beskriv en *effektiv* algoritm som skapar en tabell över alla sekvenser på *tre* respektive *fyra* tecken. Tabellen ska sorteras efter förekomst, med den vanligaste sekvensen först. Algoritmen måste beskrivas på ett strukturerat och tydligt sätt.
- b) Demonstrera hur din algoritm fungerar med ett litet exempel.
- c) Uppskatta tidskomplexiteten för din algoritm.
- d) Amöbor har särskilt långa DNA-sekvenser, med upp till  $6.7 \cdot 10^{11}$  tecken. Vi komprimerade DNA med Huffmankomprimering i uppgift E3, med mindre lyckat resultat. Föreslå en förbättring av komprimeringsalgoritmen, med inspiration från algoritmen du skrev i deluppgift a) ovan. Motivera ditt svar.