



Fredagen den 20 mars 2015 kl 9–13

Hjälpmedel: En algoritmbok och ditt eget formelblad. För betyg E krävs att alla E-uppgifter är godkända, för betyg C att alla E- och C-uppgifter är godkända, för betyg A att alla uppgifter är godkända.

1. *Lura NSA med RSA*

**Betyg E.** Rymdvarelser har uppmärksammat jordens existens och vill ta kontakt med en tildastudent på KTH. Rymdvarelserna är oroliga för att NSA ska avlyssna eller på annat sätt blanda sig i kommunikationen, men har läst att RSA-konceptet med publika och privata nycklar kan lösa kommunikationsproblematiken. Förklara vilka problem som kan lösas med hjälp av asymmetrisk kryptering med publika och privata nycklar, och berätta för rymdvarelserna hur det ska gå till.

(10 min)

2. *Trehövdade rymdvarelseers språk*

**Betyg E.** Rymdvarelserna är trehövdade och kommunicerar enbart med tre tecken ' $\oplus$ ', 'O', '=' (ett per huvud) så deras skrifter blir ganska långa. För att textsöka i skrifterna föreslår tildastudenten algoritmen KMP. Visa hur en KMP-automat för följande sökta text ser ut och ange även next-vektorn.

$\oplus O = \oplus O = = \oplus O O O$

(10 min)

3. *Bloomfilter*

**Betyg E.** Fullborda följande mening:

Ett bloomfilter har \_\_\_\_\_ krockhantering \_\_\_\_\_ genom att fylla i en eller två av luckorna med följande ordval:

- avancerad
- linjär
- kvadratisk
- logaritmisk
- ingen
- med krocklistor
- med linjär probning
- med kvadratisk probning

Det kan finnas flera lösningar men du ska bara svara med en och endast en mening. Du behöver inte fylla i båda luckorna.

(10 min)

4. *Sortera stjärnor*

**Betyg E.** Stjärnorna i rymden är olika stora, se tabellen nedan:

Stjärna	Diameter
Albireo	22 000 000 km
Aldebaran	60 000 000 km
Alpha Centauri A	1 500 000 km
Alpha Centauri B	1 000 000 km
Altair	2 600 000 km
Antares	970 000 000 km
Arctus	36 000 000 km
Betelgeuse	1 300 000 000 km

Vi vill sortera stjärnorna efter ungefärlig storlek så att alla som är ett ental miljoner km (7 siffror) hamnar först, sen tiotal miljoner km, sen hundratal miljoner km osv.

\* Visa steg för steg hur man kan få en ny sorterad vektor med avseende på storlek medelst räknearbete (distribution count).

\* Ange komplexitet för algoritmen.

(10 min)

5. *Maskhål*

**Betyg E.** När rymdresorerna reser använder de sig av maskhål (Einstein-Rosen-brygga aka wormhole) vilka förbinder två platser i rymden. De har en förteckning (graf) över vilka platser som är förbundna med varandra (det kan finnas fler än ett maskhål vid en plats). För att räkna ut en rymdfärdplan med så få maskhålsanvändningar som möjligt använder de följande algoritm tillsammans med maskhålsförteckningen samt en kö.

- 1) Lägg startmaskhålet tillsammans med en tom föräldrareferens i en kö
- 2) Plocka ut nästa maskhål och föräldrareferens ur kön och
- 3) Om detta är destinationsmaskhålet:
  - a) Skicka föräldrareferens och maskhålet till en rekursiv färdutskriftsmetod som skriver ut reshoppsvägen:
  - b) Avsluta
- 4) Annars:
  - a) Slå upp vilka andra maskhål som är förbundna med detta maskhål och lägg respektive nytt maskhål tillsammans med en föräldrareferens till detta maskhål i kön
- 5) Upprepa från punkt 2.

Rymdresorerna berättar att algoritmen fungerar ibland men ibland är det som att de bara hoppar runt mellan samma platser utan att komma fram.

\* Vad är det som är problemet med deras algoritm?

\* Förbättra algoritmen så att problemet löses!

\* Vad kallas algoritmen?

(10 min)

6. *Norrskan*

**Betyg E.** En norrskensentusiast vill lagra bilder på norrskan. Vad finns det för fördelar respektive nackdelar med en vektor respektive en länkad lista när det gäller

- a) åtkomst
- b) insättning
- c) borttagning

Svara med en tabell med tre rader.

(10 min)

7. *Planetsyntax*

**Betyg C.** En sondrobot skickar meddelanden till rymdvarelserna. En syntax används för att den ska skicka syntaktiskt korrekta meddelanden. Följande tre meningar ska godkännas:

```
Sirius, Jupiter, Venus and Pluto has minerals!  
Tellus and Alpha Centauri has life!  
Mars has minerals!
```

Ofullständiga meningar som "Venus and" ska inte godkännas.

- a) 

```
<message> ::= <planet><rest> | <message><planet><rest>  
<rest>      ::= and | has <resource>! | ,  
<planet>    ::= Sirius | Jupiter | Venus | Pluto | Tellus | Alpha Centauri  
<resource> ::= minerals | life
```
- b) 

```
<message> ::= <planet> <rest> | <planet> <rest> <planet> <rest> <resource>  
<rest>      ::= has <resource>! | , | and  
<planet>    ::= Sirius | Jupiter | Venus | Pluto | Tellus | Alpha Centauri  
<resource> ::= minerals | life
```
- c) 

```
<message> ::= <planet> has <resource>! | <planet> <rest>  
<rest>      ::= and <planet> has <resource>! | , <planet> <rest>  
<planet>    ::= Sirius | Jupiter | Venus | Pluto | Tellus | Alpha Centauri  
<resource> ::= minerals | life
```

\* Vilken eller vilka syntaxer godkänner de tre meningarna?

\* Vilken eller vilka syntaxer godkänner inte de tre meningarna - ange den mening eller de meningar som inte godkänns.

\* Vilken eller vilka syntaxer godkänner ofullständiga meningar?

(20 min)

VGW

8. *Rekursiv bränsleåtgång*

**Betyg C.** Bränsleåtgången per antal maskhålshopp som rymdvarelserna gör räknas ut med följande rekursiva metod. En tildastudent påpekar vänligt för rymdvarelserna att implementationen är onödigt ineffektiv.

```
def fuel(N):                                # N is nr of wormhole jumps
    if N == 0 :
        return 3000
    if N == 1 :
        return 2990
    else:
        return fuel(N-1) + fuel(N-2) - 3000
```

Gör följande:

- \* Ange bränsleåtgången för 6, 5, 4, 3, 2, 1 maskhålshopp
- \* Förklara vad som är ineffektivt med implementationen.

*(20 min)*

9. *Solförmörkelser*

**Betyg A.** Samtliga solförmörkelsedatum på jorden under perioden 4000 f.kr. - 4000 e.kr. är insorterade i ett binärt sökträd. Givet en historisk persons födelse och död, beskriv en algoritm som hittar antal solförmörkelser personen potentiellt skulle kunna ha upplevt under sin livstid. Algoritmbeskrivningen måste vara välstrukturerad och begriplig.

*(30 min)*

10. *Algoritm för trianglar*

**Betyg A.** Givet en förteckning över stjärnor i rymden där deras position  $x$ ,  $y$ ,  $z$  är angiven i antal ljusår:

- \* Beskriv en algoritm som räknar antalet liksidiga stjärntrianglar i rymden. Algoritmbeskrivningen måste vara välstrukturerad och begriplig.
- \* Ange din algoritms komplexitet

Stjärnavstånd kan anses lika om de är inom ett ljusårs avstånd. En tildastudent har gett dig en abstrakt funktion `distance` som returnerar avståndet mellan två rymdpositioner.

*(30 min)*