



Tisdag den 22 mars 2016 kl 14–18

Hjälpmedel: En algoritmbok (ej pythonkramaren) och ditt eget formelblad. För betyg E krävs att alla E-uppgifter är godkända, för betyg C krävs utöver detta betyg C på C-uppgifterna och för betyg A krävs högsta betyg på alla uppgifter. Lycka till!

1. *KMP*

Betyg E. Kalle Anka går under många namn. Konstruera en KMP-automat som söker efter KALLEKALLASKARL

Rita upp automaten och ange next-vektorn.

(10 min)

2. *prioritetskö*

Betyg E. Betrakta följande heap:

```
      3
     5  16
    7  8  19  20
   9 10
```

Hur ser heapen ut när man plockat ut två element? Välj det korrekta alternativet bland a) – e) nedan.

a)

```
      7
     8  8
    9 16  19  20
```

b)

```
      7
     8  16
    9 10  19  20
```

c)

```
      7
     9  8
    10 16  19  20
```

d)

```
      10
     8  16
    7  9  19  20
```

e)

```
      7
     8  9
    10 16  19  20
```

(10 min)

3. *Kö och stack*

Betyg E. Beskriv en algoritm som vänder på en kö med hjälp av en stack. Rita kön och stacken och visa hur algoritmen fungerar för kön nedan:

före vändning: Kajsa <- Kicki <- Pippi <- Titti

efter vändning: Titti <- Pippi <- Kicki <- Kajsa

(15 min)

4. *Komprimering*

Betyg E.

Vi vill komprimera meningen nedan:



PIFF PIFFAR UPP PUFF

Komprimeringen ska göras med Huffmankodning enligt den givna frekvenstabellen. Ange bitrepresentationen för varje tecken i ditt svar sorterat enligt tabellen.

| Bokstav | Frekvens | Bitrepresentation |
|---------|----------|-------------------|
| 'F' | 30 | |
| 'P' | 25 | |
| ' ' | 15 | |
| 'U' | 10 | |
| 'I' | 10 | |
| 'R' | 5 | |
| 'A' | 5 | |

(15 min)

5. *Alexander Lukas problem*

Betyg E.

Alexander Lukas letar efter ett särskilt kort i en lista av kort. Turligt nog ligger kortet han söker efter först, vilket det turligt nog alltid gör när han söker, och därför tror Alexander Lukas att sökning alltid har komplexitet 1. En tildastudent förklarar att så är det i allmänhet inte. Vad gäller i allmänhet? Avgör för varje påstående nedan om det är sant eller falskt. Att motivera svaret är frivilligt.

- Sökning i osorterad vektor har samma komplexitet som sökning i enkellänkad lista.
- Sökning i sorterad vektor är $O(\log(N))$.
- Sökning i sorterad vektor har samma komplexitet som sökning i balanserat binärträd.
- Borttagning ur en vektor är $O(N)$.

(10 min)

6. *Hamming padding*

Betyg E.

Pinocchio har lite problem med vad som är sant och falskt. Ibland stämmer inte enskilda bitar i hans berättelse. För att tolka rätt trots att bitar kan falla föreslår en tildastudent att omkoda True till 101 och False till 010.

- Vilket hammingavstånd är det mellan True och False nedan?

True 101 False 010

- Hur tolkas följande:

000 001 010 011 100 101 110 111



(10 min)

7. *Syntax för serier*

Betyg C. Vi vill ha en syntax för serietexter. Betrakta följande tre försök:

- a) <mening> ::= <subjekt><predikat><objekt>
 <predikat> ::= har
 <objekt> ::= pengar | godis | inga pengar | inget godis
 <subjekt> ::= <namn> | <namn> och <subjekt> | <namn>, <subjekt>
 <namn> ::= <förnamn> <efternamn> | <förnamn> von <efternamn>
 <förnamn> ::= Kalle | Knatte | Tjatte | Fnatte | Joakim | Musse
 <efternamn> ::= Anka | Pigg
- b) <mening> ::= <namn> har <forts>
 <forts> ::= inget <objekt> | inga <objekt> | <objekt>
 <objekt> ::= pengar | godis
 <namn> ::= Kalle Anka | Knatte, Fnatte och Tjatte Anka |
 Joakim von Anka | Musse Pigg
- c) <mening> ::= <namn> har <objekt> | <namn>, <mening> |
 <namn> och <mening> | <namn> <mening>
 <objekt> ::= pengar | godis | inga pengar | inget godis
 <namn> ::= Kalle | Anka | Knatte | Fnatte | Tjatte | Joakim von |
 Musse | Pigg

Syntaxen ska godkänna följande tre meningar:

- 1) Joakim von Anka har pengar
- 2) Knatte, Fnatte och Tjatte Anka har godis
- 3) Kalle Anka och Musse Pigg har inga pengar

Syntaxen ska underkänna följande tre meningar:

- 4) Kalle Anka har inget pengar
- 5) Fnatte von Pigg har godis

Du ska bedöma användbarheten hos syntaxerna a, b och c.

Svara med en tabell där du fyller i G om syntaxen godkänner meningen och U om den underkänner meningen. Ge även ett kort omdöme om varje syntax.

| | mening 1) | mening 2) | mening 3) | mening 4) | mening 5) |
|-----------|-----------|-----------|-----------|-----------|-----------|
| syntax a) | G | | | | |
| syntax b) | G | | | | |
| syntax c) | G | | | | |

(25 min)

8. *Oppfinnar-Jockes hashtabell*

Betyg C. En hashtabell har operationerna:

`put(key, value)` - stoppar in `key`, `value`. Om `key` finns skrivs befintligt över.

`get(key)` - returnerar `value` associerat med `key`

Man vill införa

`remove(key)` - tar bort `key` och `value` associerat med `key`

Hashtabellen är implementerad med linjär probning. Jockes första implementation av `remove` får märkliga följder. Ibland när man tar bort något så försvinner även annat man lagt in i hashtabellen. **Illustrera och förklara:**

- Vad är problemet?
- Hur löser man det?
- Skulle det varit bättre i sammanhanget om man använt kvadratisk probning?
- Skulle det varit bättre i sammanhanget om man använt krocklistor?

(20 min)

9. *Disneyland*

Betyg A. Det ska byggas ett Disneyland i Ankeborg! Där ska finnas många attraktioner som man kan gå emellan. Långben vill konstruera en algoritm som kontrollerar om man från ingången kan nå till alla attraktioner (några kanske ligger som egna onåbara circklar).

All information om vilka attraktioner som finns och vilka vägar som finns mellan attraktionerna ligger på en fil. Följande operation är given

`din_datastruktur = läs_från_fil(filnamn)`

Du får i uppgift att hjälpa Långben med algoritm och datastrukturer. För att Långben ska begripa är det viktigt att algoritmen är tydligt beskriven och att datastrukturerna även ritas upp med exempeldata.

(30 min)

10 *Rekursion*

Betyg A. En prioritetskö kan sägas vara en trädstruktur som representeras av en vektor. Konstruera en rekursiv algoritm som gör om vektorstrukturen till en trädstruktur med noder och pekare. Algoritmen ska vara $O(N)$ där N är antal element i prioritetskön. Förklara kortfattat varför din algoritm blir $O(N)$.

(25 min)

