



Måndag 24 oktober 2016 kl 14–18

Hjälpmedel: En tryckt algoritm-/datastruktursbok och ditt eget 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 på A-uppgiften (A-uppgiften kan inte kompletteras). Lycka till!

1. *KMP*

Betyg E. Spindelmannen tar inte samma vägar som andra, och lämnar vissa spår efter sig. Konstruera en KMP-automat som söker efter **VÄGGVÄGSVÄV** Rita upp automaten och ange next-vektorn.

(15 min)

2. *Trafikverkets vägdatabas*

Betyg E. Trafikverkets vägdatabas innehåller N objekt med bland annat vägens namn, vägnummer (unikt) och vägbeläggning (asfalt/grus). Vad är tidskomplexiteten för den snabbaste algoritmen som gör följande?

Motivera dina svar!

- a) Sortera vägarna i vägnummerordning.
- b) Söka efter en väg i den i vägnummerordning sorterade vektorn med *vägnumret* som nyckel.
- c) Söka efter en väg i den i vägnummerordning sorterade vektorn med *vägens namn* som nyckel. Det räcker att hitta första förekomsten av vägnamnet.
- d) Sortera vägarna efter *vägbeläggning* (asfalt/grus).

(15 min)

3. *Vägar mellan världsdelar*

Betyg E. Vissa världsdelar är förbundna med varandra. Från Europa till Asien finns det en förbindelse, likaså mellan Asien och Afrika. Det är möjligt att gå mellan Sydamerika och Nordamerika. Men den som befinner sig i Australien eller på Antarktis kan inte gå till en annan världsdel.

Rita en graf över världsdelarna och deras förbindelser, och skriv sedan upp grannmatrisen (du behöver bara fylla i förbindelserna, resten av matrisen kan lämnas tom).

Här är en lista över världsdelarna:

Afrika (AF)
Antarktis (AN)
Asien (AS)
Australien (AU)
Europa (EU)
Nordamerika (NA)
Sydamerika (SA)

(10 min)

4. *Stigs problem*

Betyg E. Stig vill hasha in 100000 ord i en hashtabell. Alla ord består av exakt 10 bokstäver. Det finns 32 olika bokstäver, och det finns en funktion $nummer(bokstav)$ som ger varje bokstav ett nummer mellan 1 och 32.

Hjälp Stig genom att förklara varför den ena av följande två hashfunktioner är bättre än den andra. Du får anta att tabellens storlek är väl vald, och att krockhanteringen fungerar bra. Använd överslagsberäkningar och rita gärna.

- summan av bokstävernas nummer modulo tabellens storlek
- produkten av bokstävernas nummer modulo tabellens storlek

(15 min)

5. *Vägbeskrivning*

Betyg E. Anta att vi vill komprimera meningen nedan med Huffmankodning. Vi samlar in statistik enbart från de 26 tecknen i meningen. Mellanslag räknas som ett tecken.

ÅK E4 SEN E18 MOT ENKÖPING

- a) Ange två tecken som får korta bitrepresentationer.
- b) Ange två tecken som får långa bitrepresentationer.

Du behöver inte genomföra Huffmankodningen, men du måste motivera dina svar.

(10 min)

6. *Syntax för sökväg*

Betyg E. Här är en syntax för sökvägar:

```
<SÖKVÄG> ::= <ENHET> ":" | <ENHET> ":" <LISTA>
<LISTA>  ::= "\"<NAMN> | "\"<NAMN> <LISTA> | <NAMN>
<ENHET> ::= "C" | "D" | "E" | "H"
<NAMN>  ::= [a-z]*
```

Vilka av följande sökvägar godkänns av syntaxen? Motivera dina svar!

- a) C:\labbar\
- b) D:\2016\labbar
- c) E:
- d) H:\tilda\labbar

(15 min)

7. *Vägen genom en graf*

Betyg C. Breddenförstsökning, djupetförstsökning och bästaförstsökning är tre olika sätt att hitta en väg genom en graf.

Men vad är skillnaden? Beskriv de tre algoritmerna och formulera för varje algoritm ett problem som bäst löses med just den algoritmen (och där de två övriga inte är lika bra).

(30 min)

8. *Vägningsar*

Betyg C. Det går ett rykte om att tyngre Pokémon har starkare attacker.

Tilda vet inte vad någon av hennes Pokémon väger och har bara en balansvåg (som visar vilken av två Pokémon som är tyngst men inte den exakta vikten).

Tilda vill nu sortera sina Pokémon i viktordning. Du erbjuder tre sätt att göra sortering, med hjälp av de tre olika datastrukturerna nedan:

- En prioritetsskö
- Två stackar
- Ett binärt sökträd

Visa med exempel hur de tre olika sorteringarna fungerar. Rita!

Resonera sedan om effektiviteten (jämförelser och förflyttningar) för respektive sortering. Spelar det till exempel roll om Tilda har många tunga Snorlax som är svårflyttade? Eller många Zubat som är svåra att väga?

(30 min)



9. *Vägen mellan två ord*

Betyg A. Levenshteinavståndet är det kortaste redigeringsavståndet mellan två strängar. Enligt Wikipedia: "Avståndet beräknas som summan av det antal raderingar, infogningar och substitueringar av tecken som krävs för att transformera den ena strängen till den andra." Levenshteinavståndet blir aldrig större än längden av den längsta av strängarna.

Exempel:

al
ål

En substituering, alltså avstånd 1 mellan al och ål.

strand
trand
rand
and

Tre raderingar, alltså avstånd 3 mellan strand och and.

plågoris
alågoris
algoris
algorit
algoritm

Två raderingar, en substitution och en infogning, alltså avstånd 4 mellan plågoris och algoritm.

- Konstruera en rekursiv algoritm för att beräkna Levenshteinavståndet mellan två ord. Algoritmen ska inte skriva ut hur redigeringen går till, utan bara returnera avståndet. Din algoritmbeskrivning måste vara så tydlig att den går att programmera.
- Visa hur din algoritm fungerar genom att demonstrera ett par av de rekursiva anropen för vart och ett av de tre exemplen ovan.
- Vilken tidskomplexitet har din algoritm, är den logaritmisk, linjär, polynomisk eller exponentiell? Motivera ditt svar.

(40 min)