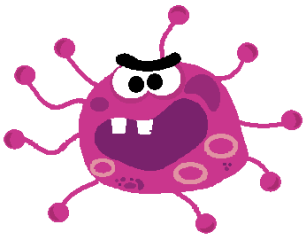
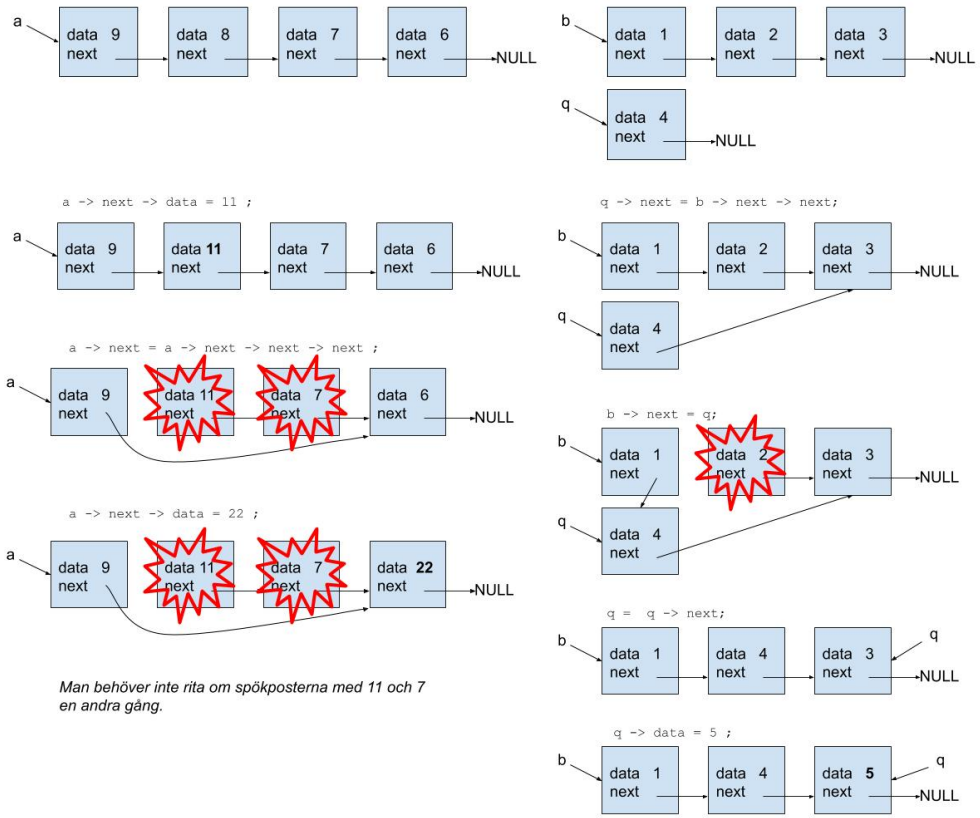
KTH EECS, Alexander Baltatzis



DD1321 TENTAMEN I
TILLÄMPAD PROGRAMMERING OCH DATALOGI
Onsdag 11 mars kl 14–18

E 0.



Man behöver inte rita om spökposterna med 11 och 7 en andra gång.

E 1. *KMP*

Vi söker efter virala virus som virvlar omkring i världen.

15 min Rita en KMP-automat för ordet *VIRALAVIRVELVIRUS* samt ange next-vektorn.

```
V I R A L A V I R V E L V I R U S
0 1 1 1 1 1 0 1 1 4 2 1 0 1 1 4 1
```

E 2. *Bubbelsortering*

15 min

a) Följande sex tal ska sorteras i stigande ordning med bubbelsortering. Visa i vilken ordning talen ligger för varje varv i sorteringen.

```
8, 1, 7, 9, 2, 4
```

b) Hur många varv kommer bubbelsorteringen att köra för detta exempel? Motivera kort.

```
varv 0) 8 1 7 9 2 4
varv 1) 1 7 8 2 4 9
varv 2) 1 7 2 4 8 9
varv 3) 1 2 4 7 8 9

b) 4 varv (ett extra där inga byten sker) även alla varv godkänns
```

E 3. *Graf*

En vanligt mekanism för uppkomst av nya virus är genom "Cross-species transmission"(CST) där en art överför ett virus till en annan. Givet nedan är några förekommande sådana där en riskfaktor (1-3, lågt till högt) anges.

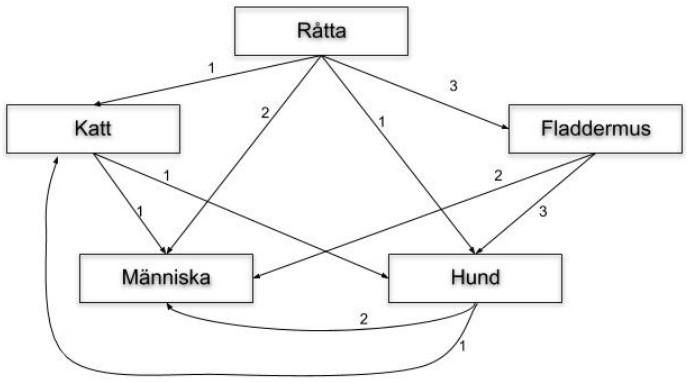
```
Råtta -> Fladdermus (3)
Råtta -> Människa (2)
Råtta -> Hund (1)
Råtta -> Katt (1)
Fladdermus -> Människa (2)
Fladdermus -> Hund (3)
Hund -> Katt (1)
Hund -> Människa (2)
Katt -> Människa (1)
Katt -> Hund(1)
```

15 min

a) Rita en graf (med hörn och kanter) som beskriver dessa överföringar.

b) Hur många hörn har grafen? Svar 5

c) Hur många kanter har grafen? Svar 10



E 4. *Kryptering*

Du behöver skicka ett meddelande till Dr Mildred Who men denne är rädd för att det kan innehålla virus eller bli infekterat på vägen. Du bestämmer dig därför för att använda kryptering och verifiering med RSA.

Det okodade meddelandet ligger i variabeln `message`.

Du har tillgång till funktionen `rsa` som tar en nyckel och en text och returnerar en kodad text. Nu vill du skriva ett program för att skicka `message`.

10 min *Vilket eller vilka* av alternativen nedan kan du använda för att skicka meddelandet så att enbart Dr Who kan dekryptera det och vara säker på att det är du som skickat det.

a)
```
text2 = rsa ( MinPrivataNyckel, message )
text3 = rsa ( DrWhoPrivataNyckel, text2 )
mail ( "DrMildredWho@who.org", text3 )
```

b)
```
text2 = rsa ( MinPublikaNyckel, message )
text3 = rsa ( DrWhoPublikaNyckel, text2 )
mail ( "DrMildredWho@who.org", text3 )
```

c)
```
text2 = rsa ( MinPrivataNyckel, message )
text3 = rsa ( DrWhoPublikaNyckel, text2 )
mail ( "DrMildredWho@who.org", text3 )
```

d)
```
text2 = rsa ( MinPublikaNyckel, message )
text3 = rsa ( DrWhoPrivataNyckel, text2 )
mail ( "DrMildredWho@who.org", text3 )
```

e)
```
text2 = rsa ( DrWhoPublikaNyckel, message )
text3 = rsa ( MinPrivataNyckel, text2 )
mail ( "DrMildredWho@who.org", text3 )
```

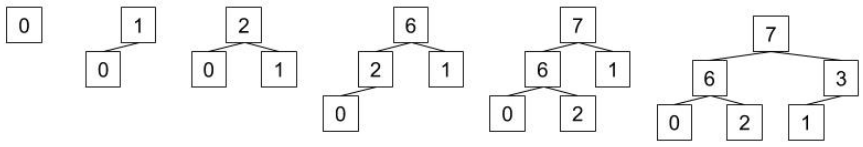
rätt svar c) och e) Man kan göra kryptering och signering i olika ordning.

E 5. *Heap*

För att rangordna olika hotbilder mot grundläggande samhällsfunktioner (t e x cyberattacker, miljökatastrofer eller pandemier) använder myndigheten för samhällsskydd och beredskap (MSB) sig av en max-heap där varje potentiell samhällsfara har en riskfaktor (0-10, låg till hög). I kronologisk ordning inträffar följande sex händelser som ska läggas in i denna heap.

Rita detta steg för steg (det räcker att du anger riskfaktorn). Du kan välja mellan
10 min att rita heapen på trädform eller på vektorform.

```
Generaldirektörens guldfisk dör: 0
Jordbävning: 1
Zombieapokalyps: 2
Pandemi: 6
Mello lägger ner: 7
Soleruption: 3
```



E 6. *Komprimering och felkorrigering*

Ett s k DNA-virus innehåller genetiska sekvenser med nukleotiderna adenin (A), guanin (G), cytosin (C) och tymin (T) vilket skrives t ex CGCCTATACGGA. I en databas med över 10^{12} genetiska sekvenser lagras de *fyra* tecknen CGAT med ASCII-kod som kräver sju bitar per tecken. För att spara plats vill man istället lagra på kompaktast möjliga vis. Vi förutsätter här att varje nukleotid är *lika vanligt förekommande*. Motivera dina svar!

10 min

a) Hur skulle kunna denna kodning kunna se ut? Visa med ett exempel.

b) För att upptäcka bitfel vid läsning av sekvenser vill man feldetektera med en *paritetsbit*. Förklara hur detta skulle kunna gå till, och visa med ett exempel.

Man kan koda tecknen med två bitar var t.ex. C=00 G=01 A=10 T=11. Man kommer fram till samma svar med huffmanalgoritmen eftersom tecknen är lika vanligt förekommande. Man kan lägga till en paritetsbit på slutet av varje tecken C=001 G=011 A=101 T=111

C 7. *Bloomfilter* Ett antal viruspatienter har satts i karantän. Karantänen för den enskilde varar som mest i 14 dagar och man är beredd på en hög omsättning patienter. Det finns ett maxtak på hur många patienter man kan husera samtidigt. Man vet inte hur länge epidemin varar. Viruspatienterna behöver tillfälliga personliga lösenord. För att inte lagra lösenorden i klartext så tänker man använda bloomfilter. Valet står mellan att lagra lösenorden i ett *bloomfilter* eller *hashade och saltade*. Jämför dessa två alternativ. Var noga med att motivera dina slutsatser med hänvisning till de givna förutsättningarna.

25 min

Man kan inte ta bort ett lösenord från bloomfiltret. Nollställer man ett ords 14 ettor så påverkar man andra ord som använder samma ettor. Det går inte heller att bygga om bloomfiltret eftersom lösenorden inte finns i klartext någon annanstans. Den höga omsättningen kommer leder till vektorn fylls utav ettor och mängden false positives kommer att öka. Det går inte att utnyttja att det finns ett maxtak på antal samtidiga patienter eftersom man inte kan ta bort de patienter som skrivs ut. Om man bara lagrar lösenorden i bloomfiltret finns dessutom ingen koppling mellan användare och lösenord så man kan logga in med andra användares lösenord. Visserligen lagras inte lösenorden i klartext men sammantaget är bloomfilter ingen bra lösning givet förutsättningarna.

Om man lagrar patient och lösenord hashade och saltade på fil så lagras de inte i klartext. Det går det att ta bort patienter varefter de skrivs ut oavsett omsättning och man kan utnyttja att det bara är ett visst antal samtidiga patienter inlagda.

A 8. Givet ett binärt sökträd som innehåller heltal.

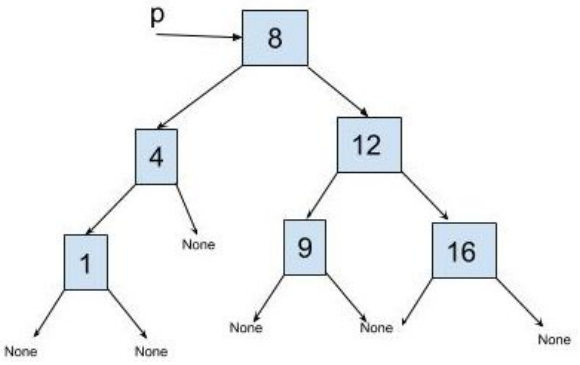
a) Konstruera en effektiv algoritm som givet (1) en pekare till ett binärt sökträd och (2) en sökt summa, returnerar två nodpekare. Summan av värdena i de två noderna ska bli den sökta summan. Om det inte går att hitta två sådana noder så ska *None, None* returneras.

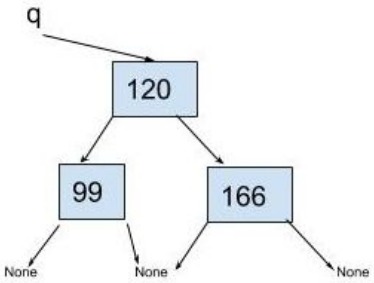
25 min Algoritmen och datastrukturer ska vara tydligt beskrivna.

b) Ange komplexitet för din algoritm.

c) Visa hur din algoritm fungerar om man söker efter summan 10 i trädet p nedan.

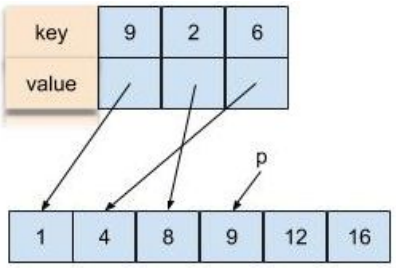
d) Visa hur din algoritm fungerar om man söker efter summan 10 i trädet q nedan.



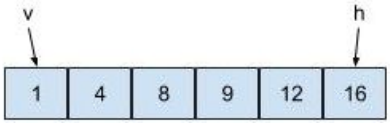


Det går att lösa problemet i O(N) för betyg A. Om man går igenom trädet inorder kan man bygga upp en sorterad lista/vektor istället vilket kan vara enklare för att förklara algoritmen.

Låt en pekare p gå igenom listan, för varje nod kolla om dess värde finns i en dictionary/hashtabell. Om inte, lägg in (summan - värde) som key och pekaren till noden som value. Första elementet 1 finns inte i hashtabellen så lägg in 9 som pekar på noden med 1.



Efter tre varv ligger 9, 6 och 2 i hashtabellen och då är man framme på noden med värdet nio. Eftersom 9 finns i hashtabellen så är man klar och returnerar p och pekaren till 1. Man kan göra samma algoritm genom att gå igenom trädet direkt i in-, pre- eller postorder (se suboptimeringar) och lägga in de matchande talen i dictionary allt eftersom.



En annan lösning i O(N) är att använda två pekare h, v som börjar i varsin ände. Låt högerpekaren stega sig inåt så länge som summan är större än den sökta. När summan är mindre stegar man vänsterpekaren. Genom att stega varsamt ömsom högerpekare och vänsterpekaren i små steg fram och tillbaka kan man ringa in den sökta summan i O(N). Om man inte är varsam kan det resultera i en kvadratiskt lösning där man för varje högerpekare stegar igenom alla vänsterpekare. Det är mycket lättare att beskriva traverseringen för den här algoritmlösningen på en lista/vektor än på ett träd.

Det går också att lösa problemet i O(NlogN) för betyg B. Man traverserar trädet i in- pre- eller postorder och binärsöker i trädet efter den sökta summan minus nodens värde.

Det går att göra en del suboptimeringar som att t.ex. inte undersöka tal som är betydligt större än den sökta summan. Det är lätt att göra fel och kapa trädet utan att undersöka vänsternoder som skulle kunna vara inom intervallet. Om det finns negativa tal i trädet så behöver man veta trädets minsta tal för att avgöra intervallet man letar inom. Då kan det vara bra att gå igenom trädet inorder och börja med minsta noden.