



**DD1320/DD1321/DD1325**  
**TENTAMEN I TILLÄMPAD DATALOGI**  
**Måndag 21 oktober 2019 kl 14–18**

Hjälpmedel: Egenskrivet formelblad, max 4 sidor (eller två dubbelsidiga A4). För betyg E krävs att alla E-uppgifter är godkända (upp till två E-uppgifter kan kompletteras). För betyg D respektive C krävs (utöver E-kraven) D respektive C på C-uppgiften. För betyg B respektive A krävs (utöver C-kraven) betyg B respektive A på A-uppgiften. Lycka till!

**E 1.** *KMP*

Den som vill få fler sökträffar på sitt företags webbsida kan t ex kopiera en konkurrents webbsida och klistra in den på sin egen. Men det bör man inte göra - det är inte en legitim strategi.

15 min

Rita en KMP-automat för ordet ILLEGITIM samt ange next-vektorn.

**E 2.** *Tidskomplexitet*

Ville Vessla har testat följande fyra algoritmer, men glömde skriva upp vilken rad som motsvarar vilken algoritm.

- a) Urvalssortering av  $n$  element
- b) Linjärsökning bland  $n$  element
- c) Binärsökning bland  $n$  element
- d) Sökning i hashtabell med  $n$  element

15 min

Tabellen nedan innehåller körtider för fyra olika  $n$ . Vilken algoritm ska stå på vilken rad, och vilken tidskomplexitet har algoritmerna?

	Algoritm	$n = 10000$	$n = 20000$	$n = 40000$	$n = 80000$	Tidskomplexitet
1		0.0008	0.0016	0.0032	0.0064	
2		$4.4 \cdot 10^{-6}$	$4.4 \cdot 10^{-6}$	$4.3 \cdot 10^{-6}$	$4.4 \cdot 10^{-6}$	
3		$1.0 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$	$1.4 \cdot 10^{-5}$	$1.6 \cdot 10^{-5}$	
4		15	60	240	960	

**E 3.** *Komprimering***Betyg E.**

I Rövarspråket är "o" mycket vanligare än alla andra bokstäver. (Varje konsonant dubblas, med o emellan, t ex skulle "abc" bli "abobcoc" på rövarspråket.)

Fyra av rövorna (Agata, Bobo, Cecil och Dorste) har beräknat Huffmankoder för den givna frekvenstabellen, men alla har fått olika svar.

15 min

Vem har rätt? Avgör det genom att rita Huffmanträdet.

Bokstav	Frekvens i procent	Agata	Bobo	Cecil	Dorste
O	30	11	00	01	00
B	16	10	10	110	010
D	14	01	010	101	0110
N	14	00	011	010	01110
T	13	111	110	011	011110
A	7	110	1110	1000	0111110
I	6	101	1111	1100	01111111

**E 4.** *Kryptering*

Stadens maffiabossar behöver hjälp med kryptering.

15 min

- a) Vilken av dessa två krypteringsmetoder skulle du rekommendera, Caesarchiffer eller One-time pad? Motivera ditt svar.
- b) Förklara steg för steg hur Alice Capone ska göra för att skicka ett krypterat brev till Bob "Bugs" Malone. Använd metoden du valde ovan.
- c) Illustrera själva krypteringen med ett konkret exempel.

**E 5.** *Heap*

Juvel-ligan samlas efter rånet för att dela upp bytet mellan sig i hierarkisk ordning. Chefen får det dyraste smycket, kassaskåpsknäckaren det näst dyraste, osv (prao-eleven får nöja sig med det som blir kvar till sist). För enkelhets skull lägger dom in smyckenas värden i en max-heap.

15 min

Hur ser heapvektorn ut när man satt in värdena:

4200, 1900, 7300, 5800, 8400, 1400, 4500 ?

Välj bland alternativen nedan (som visar vektorformen). Motivera ditt svar genom att rita, steg för steg, hur heapen byggs upp.

- a) [8400, 7300, 5800, 4500, 4200, 1900, 1400]
- b) [4500, 7300, 1900, 8400, 5800, 4200, 1400]
- c) [4500, 1900, 7300, 1400, 4200, 5800, 8400]
- d) [8400, 4500, 7300, 1900, 5800, 1400, 4200]
- e) [8400, 7300, 4500, 1900, 5800, 1400, 4200]
- f) [8400, 4500, 7300, 1400, 1900, 5800, 4200]

**E 6.** *Reguljärt uttryck*

Hemliga epostadresser på Darknet får bara innehålla bokstäver, siffror och tecknen "@" samt "." (punkt).

En epostadress består av en *namndel*, följt av tecknet "@", följt av en *domändel*. Namndelen får innehålla godtyckligt antal bokstäver, siffror och punkter, i valfri ordning. Domändelen består av ett godtyckligt antal bokstäver, följt av **.dnt**.

15 min

Skriv ett reguljärt uttryck som matchar en sådan epostadress, t ex

*yegg23@dacoit.dnt*

Några regler för reguljära uttryck:

<p><math>a^*</math> matchar noll eller flera <math>a:n</math> <math>a^+</math> matchar ett eller flera <math>a:n</math> <math>a?</math> matchar ett eller inget <math>a</math> <math>.</math> matchar alla tecken utom radslut <math>[a-zA-Z]</math> matchar alla engelska bokstäver <math>[abc]</math> matchar <math>a</math>, <math>b</math> eller <math>c</math> <math>[\^abc]</math> matchar vilket tecken som helst utom <math>a</math>, <math>b</math> eller <math>c</math> <math>X Y</math> matchar uttrycket <math>X</math> eller uttrycket <math>Y</math> <math>\.</math> matchar en punkt. <math>(ab)^+</math> skapar en grupp. <math>T</math> ex matchar <math>(ab)^+</math> ett eller flera <math>ab:n</math></p>
---

**C 7.** *Datastruktur för illegala primtal*

Tilda har startat företaget *Prima Primtal* som erbjuder sina kunder garanterat prima tal som kan användas t ex vid kryptering. Men ett problem är att en del länder förbjuder innehav och distribution av vissa så kallade "illegala primtal". Hon måste därför kontrollera att de primtal hon erbjuder sina kunder inte är illegala, utan att själv åka fast för innehav.

För att hålla reda på de illegala primtalen väljer hon mellan följande datastrukturer:

- binärt sökträd
- bloomfilter
- hashtabell
- prioritetsskö implementerad med en heap

*35 min*

För *var och en* av de fyra datastrukturerna ovan ska du:

- a) Beskriva i detalj hur datastrukturen kan användas för att hålla reda på de illegala primtalen.
- b) Motivera varför den är lämplig/olämplig för att hålla reda på dessa primtal.

**A 8.** *Algoritm för mutor*

**Betyg A.**

Det finns  $n$  tjänstemän på myndigheten och du måste muta dem alla för att dom ska se genom fingrarna när det gäller dina illegala aktiviteter. En tjänsteman låter sig mutas om mutan är tillräckligt stor, men varje tjänsteman har sin egen gräns för när denne accepterar en muta. Du kan anta att det finns en metod  $acceptera(m)$  som returnerar True om mutan (som är  $m$  kronor) är tillräckligt stor (för just denna tjänsteman), och False annars.

*55 min*

- a) Beskriv en *effektiv* algoritm som avgör om du har råd att muta alla de  $n$  tjänstemännen. Du kan anta att du har totalt  $x$  kronor. Beskriv algoritmen på ett tydligt och strukturerat sätt.
- b) Demonstrera hur din algoritm fungerar med ett litet exempel.
- c) Uppskatta tidskomplexiteten för din algoritm, och visa hur du kom fram till den.