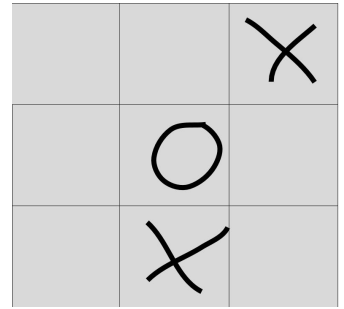


DD1320/DD1325
TENTAMEN I TILLÄMPAD DATALOGI
Måndag 16 december 2019 kl 8–12



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*

Redan de gamla egyptierna spelade tripp-trapp-trull (tre-i-rad).

15 min

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

E 2. *Binärt sökträd*

Brädspelscaféet har just fått leverans av nya spel. Hjälp Zoylee att sortera in spelen i ett binärt sökträd.

Imhotep, Team3, Squadro, Alhambra, Dropit, Kittycat, MrJack, Quacks, Welcome

15 min

- a) Sortera in spelen i tur och ordning (först Imhotep, sen Team3, osv) i ett binärt sökträd. Det är bokstavsordning som gäller. Det räcker att du anger första bokstaven.
- b) I vilken ordning kommer spelen om du skriver ut i preorder?
- c) I vilken ordning kommer spelen om du skriver ut i inorder?
- d) I vilken ordning kommer spelen om du skriver ut i postorder?

E 3. *Komprimering*

Den som spelar Overwatch trycker oftare på vissa tangenter. Om vi skulle komprimera en spelares tangenttryckningar skulle därför Huffmankodning passa bra. Nedan finns en tabell som visar hur ofta en viss tangent används.

15 min

- a) Rita Huffmanträdet.
- b) Ange bitrepresentationen för varje tecken i tabellen.

Bokstav	Frekvens i procent
A	25
D	25
W	20
S	15
R	10
TAB	3
ESC	2

E 4. *Kryptering*

I filmen *The Imitation Game* försöker Alan Turing hitta ett sätt att knäcka kryptering. Inspirerad av filmen vill du nu göra samma sak.

10 min

Du har fått ett krypterat meddelande av en kurskamrat från Tilda-kursen, men du känner inte till på vilket sätt det är krypterat.

- a) Föreslå en metod för att knäcka krypteringen.
- b) Ange en krypteringsmetod där ditt förslag i a) inte kommer att fungera. Berätta varför!

E 5. *Heap*

För att hålla reda på det mest populära spelet (räknat i miljoner spelare) är det praktiskt att använda en max-heap.

15 min

Hur ser heapvektorn ut när man satt in värdena nedan?

81, 75, 34, 88, 19, 64, 74

Börja med att lägga in 81 i heapen.

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

- a) [88, 75, 74, 81, 19, 34, 64]
- b) [19, 64, 34, 88, 74, 75, 81]
- c) [88, 81, 75, 74, 19, 34, 64]
- d) [19, 64, 34, 74, 81, 75, 88]
- e) [88, 81, 74, 75, 19, 34, 64]
- f) [19, 34, 64, 74, 75, 81, 88]

E 6. *Bloomfilter*

När du loggar in på ditt konto på spelsajten Steam så funderar du över hur ditt användarnamn och lösenord är lagrade. Du vet ju att det är olämpligt att lagra inloggningsuppgifter i klartext.

10 min

Det skulle kunna vara ett Bloomfilter som används för att kontrollera att lösenordet är korrekt. Men som du säkert minns kan Bloomfilter ge fel med en viss (liten) sannolikhet. Vad skulle detta ha för konsekvenser här?

- a) ett felaktigt lösenord kan accepteras
- b) ett korrekt lösenord kan nekas
- c) både a) och b) ovan kan inträffa

Motivera ditt svar. Rita gärna!

C 7. *Jämföra datastrukturer*

Tänk dig ett spel där man ska kunna resa mellan olika städer. En direktresa från en stad till en annan är möjlig om det finns en järnvägslinje mellan de två städerna. En stad kan ha flera järnvägslinjer.

Alla städer, och förbindelserna emellan dem behöver lagras i en datastruktur. Den ska kunna användas för att avgöra om en direktresa från en stad till en annan är möjlig.

Valet står mellan att använda:

- pekare för att ange till vilken stad en järnvägslinje leder
- index för att ange till vilken stad en järnvägslinje leder

45 min

- a) Rita och beskriv med ett exempel hur en datastruktur kan byggas upp där *pekare* används.
- b) Rita och beskriv med ett exempel hur en datastruktur kan byggas upp där *index* används.
- c) Ange två egenskaper som är intressanta att jämföra för dessa datastrukturer.
- d) Gör en jämförelse mellan datastrukturerna och presentera dina resultat på ett strukturerat sätt.

A 8. *Algoritm*

Uppvärmning:

55 min

- a) Beskriv en enkel algoritm för att hitta det största och det nästa största elementet i en lista med $n = 2^k$ element.
- b) Demonstrera hur din algoritm fungerar med ett litet exempel.
- c) Beräkna antalet jämförelser för din algoritm. Visa hur du kom fram till svaret.

Samma problem kan formuleras på följande sätt:

Anta att vi ska ordna en Hearthstone-turnering med $n = 2^k$ spelare, där de två *bästa* ("största" och "näst största") spelarna får pris. Vi vill ha så få matcher som möjligt, så den enkla algoritmen du skrev ovan duger inte.

- d) Beskriv en *effektiv* algoritm med $n + k - 2$ matcher som hittar den bästa och den näst bästa spelaren i turneringen. Beskriv algoritmen på ett tydligt och strukturerat sätt.
- e) Demonstrera hur din algoritm fungerar med ett litet exempel, där $n = 8$.
- f) Visa att antal matcher blir $n + k - 2$ för din algoritm.