

Tentamen, 2.5 högskolepoäng

November 13, 2024, 8:15 - 12.15

Du ska inte använda någon bok, bilder eller online-resurs. Du kan använda en fickräknare. Du får inte samarbeta med andra studenter via mobiltelefon eller chatt eller att prata. Misstänkta fall av samarbete och plagiering kommer att rapporteras och tentan anses ogiltig.

Du kan svara på svenska eller engelska, men engelska är att föredra. Oroa dig inte om din engelska inte är perfekt, den kommer inte att utvärderas, men svaren måste vara förståeliga.

*Tentamen har 10 frågor med en sammanlagd poäng på 25. Krav för betyg 3 (godkänt) är **15** poäng, krav för betyg 4 är **19** poäng och krav för betyg 5 är **23** poäng.*

Du borde svara med papper och penna. Se till att skriva numret på den fråga du svarar på. Svar som inte kan tolkas på grund av dålig handskrift eller dålig grammatik ger inga poäng.

För frågor, du kan ringa Dario till 0723756073.

Tips: läs igenom HELA tentamen först och lägg upp en strategi för i vilken ordning ni löser uppgifterna – tex de ni tycker är enkla först och de svåra i mån av tid.

Fråga 1 (2pt)

- Vad är den största skillnaden mellan en vanlig (icke-cirkulär) länkad lista och en cirkulär (eng: circular) länkad lista? (1pt)
- Är det korrekt att en kö (eng: queue) är en vanlig datastruktur för att implementera en buffert (eng: buffer)? (0.5 pt)
- Stämmer det att en stack är en FIFO (first-in-first-out) datastruktur? (0.5pt)

Svar.

- I en vanlig (icke-cirkulär) länkad lista pekar sista elementet på null (typiskt), medans i en cirkulär länkad lista pekar sista elementet tillbaka till det första elementet.
- Ja, det är korrekt.
- Nej, det stämmer inte.

Fråga 2 (2pt)

Rita ett tillståndsdigram för en finit tillståndsmaskin (FSM) som består av minst 3 tillstånd (states), 4 tillståndsoverföringar (state transitions), samt villkor för tillståndsoverföringar (conditions for transitions).

Svar:

Se föreläsningsanteckningar.

Fråga 3 (3pt)

Beakta följande funktioner som alla beror på n .

- $f(n) = n^2 + n \cdot \log_2 n$
- $g(n) = 4n^4 + n$
- $h(n) = 3n + 4 \cdot \log_2 n$
- $i(n) = n^2 + 50n + 4$
- $j(n) = n^3 + 2n^2$

- a. Vilken (eller vilka) av funktionerna är $\theta(n)$? Motivera kortfattat ditt svar. (1pt)
- b. Vilken (eller vilka) av funktionerna är $\theta(n^2)$? Motivera kortfattat ditt svar. (1pt)
- c. Vilken (eller vilka) av funktionerna är $\theta(n^3)$? Motivera kortfattat ditt svar. (1pt)

Svar:

- a. $h(n)$. The growth rate of this function is proportional to n .
- b. $f(n), i(n)$. The growth rate of these functions is proportional to n^2 .
- c. $j(n)$. The growth rate of this function is proportional to n^3 .

Fråga 4 (3p)

Den här frågan testar er kunskap om algoritmen *merge sort*.

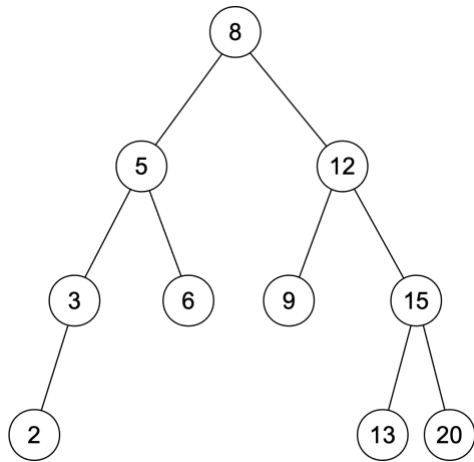
- a. Vad har *merge sort* för tidskomplexitet? Använd θ -notation. (1pt)
- b. Hur många jämförelser krävs i värsta fall för att sammanfoga (eng: merge) 2 listor med vardera 4 element, det vill säga totalt 8 element? Ge ett exempel (på två listor med 4 värden var) när värsta-fallet inträffar? (2pt)

Svar:

- a. Merge sort har tidskomplexiteten $\theta(n \log n)$.
- b. 1p för rätt svar, och 1p för korrekt förklaring av när värstafallet inträffar. Det behövs i värsta fall 7 jämförelser, och detta fall inträffar när största elementet av de 8 är i ena listan och det näst största i den andra listan. Tex lista 1 innehåller elementen 1, 2, 3, 8 och lista 2 innehåller elementen 4, 5, 6, 7.

Fråga 5 (3p)

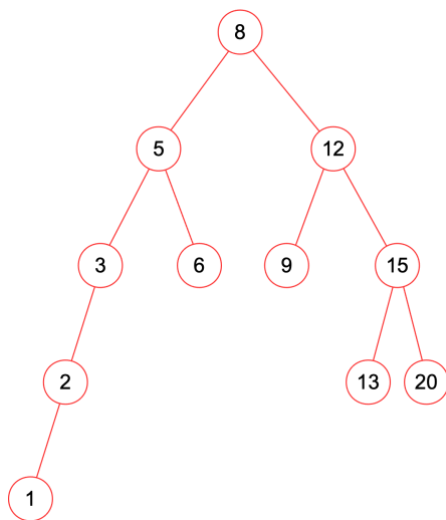
- a. Är följande binära sökträd balanserat? (1pt)



Svar: Trädet är balanserat.

- b. Lägg till en nod med värde 1 i sökträdet i trädet ovan, och visa hur trädet ser ut efteråt. Är trädet balanserat? Motivera kortfattat ditt svar. (2pt)

Svar:



Trädet är inte balanserat. Höjden på vänster och höger subträd skiljer sig mer än ett, till exempel för det subträd som har nod 3 som rot.

Fråga 6 (2p)

EN: Describe the steps needed to compile a piece of code from source to executable.

SE: Beskriv stegen som behövs för att kompilera en bit kod från källa till körbar fil.

Answer:

1. Preprocessing: performs simple text modification.
2. Compilation: generates assembler code from source code.
3. Assembly: generates machine instructions from assembler code.
4. Linking: links all “modules” together into one executable.

Listing the steps without any description gives 1pt.

Fråga 7 (2p)

EN: Provide examples of sleep modes on a microcontroller.

SE: Ge exempel på vilolägen (*sleep modes*) på en mikrokontroller.

Answer:

- Active: no sleep, everything is ON
- Idle: processor is off, but RAM and timers are on
- Deep sleep: turns off also peripherals, except those that can generate interrupt
- Deep hibernation: leave RAM unstable but registers are kept
- Power off: needs a complete reboot

Fråga 8 (2p)

EN:

Describe the use of the **stack** and the **heap** memory in C. (2pt)

Hints: How and when they are allocated? What do they contain?

SE:

Beskriv användningen av **stacken** och **heapminnet** i C.

Tips: Hur och när de tilldelas? Vad innehåller de?

Answer:

Stack memory is used to allocate temporary memory when a function is called. Each time the function is called, the stack is allocated a piece of memory needed to store a copy of the arguments, local variables and the return address.

Heap memory is allocated manually by the programmer. It is used for variables the life of which is limited in time (otherwise they would be allocated on the static memory) and need to survive after the execution of a function.

Fråga 9 (4p)

EN

You are building a digital thermometer to be used on cats like this one:



A resistive sensor that measures temperature is connected to the 10-bit ADC of a microcontroller through a pullup resistor of 500 Ohm. The datasheet of the sensor specifies that the resistance of the sensor changes depending on the temperature T (in Celsius degrees) placed on it according to the formula: $R = K \log(T)$ where K is a constant.

Derive the formula that links gives the temperature, in Celsius degrees, from the output of the ADC.

Hint: if you don't remember the formula from the lecture, use the voltage divider formula ($V_{out} = V_{cc} R_2 / (R_1 + R_2)$) and assume that the value produced by the ADC is proportional to the voltage ($A = \frac{V_{out}}{V_{cc}} A_{max}$). You know A_{max} from the resolution of the ADC.

SE

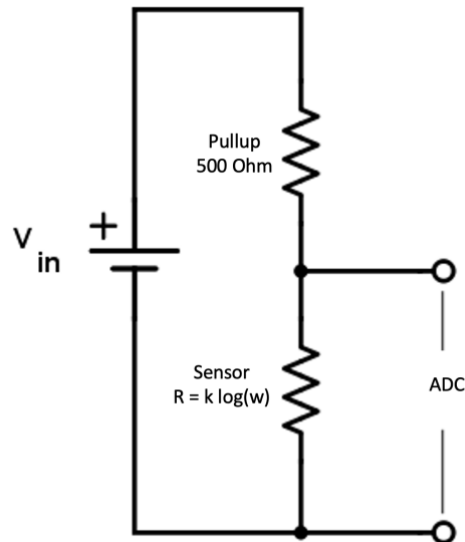
Du bygger en digital termometer som ska användas på katter som den här:



En resistiv sensor som mäter temperatur är ansluten till 10-bitars ADC på en mikrokontroller genom ett pullup-motstånd på 500 Ohm. Sensorns datablad anger att sensorns resistans ändras beroende på temperaturen T (i Celsiusgrader) placerad på den enligt formeln: $R = K \log(T)$ där K är en konstant.

Härled formeln som länkar ger temperaturen, i Celsiusgrader, från utsignalen från ADC.

Tip: om du inte kommer ihåg formeln från föreläsningen, använd spänningsdelarformeln ($V_{out} = V_{cc} R_2 / (R_1 + R_2)$) och antag att värdet som produceras av ADC:n är proportionellt mot spänningen ($A = \frac{V_{out}}{V_{cc}} A_{max}$). Du känner A_{max} från ADC:ns upplösning.



ANSWER:

Resolving the voltage divider and the proportionality of the ADC output (see slides for details), we get that the resistance of the sensor is $R = \frac{A \cdot 500}{1024 - A}$. From $R = K \log(T)$ we derive

that $w = 10^{R/T}$ therefore: $T = 10^{\frac{A \cdot 500}{1024 - A} / K}$

Fråga 10 (2pt):

EN: Show how UML diagrams can be used for the 4+1 view architectural model. Fill in the following table by placing the UML diagrams from the list below into the right column.

SE: Visa hur UML-diagram kan användas för arkitekturmodellen med 4+1-vy. Fyll i följande tabell genom att placera UML-diagrammen från listan (se ner) i den rätta kolumnen.

View	Logical view	Process view	Development view	Physical view	Scenarios
UML – Diagram					

UML diagrams:

Class Diagram, Object Diagram, Activity Diagram, State Diagram, Timing Diagram, Sequence Diagram, Component diagram, Package diagram, Deployment diagram, Use case diagram.

ANSWER:

View	Logical view	Process view	Development view	Physical view	Scenarios
UML – Diagram	Class, Object Diagram	Activity, State, Timing, Sequence Diagram	Component, Package diagram	Deployment diagram	Use case diagram