

Tentamen, 2.5 högskolepoäng

January 16, 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 9 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 (3pt)

Beakta följande funktioner som alla beror på n .

- $f(n) = 400n^2 + 3\log_2 n$
- $g(n) = 8n^3 + 2n$
- $h(n) = n + 2$
- $i(n) = n^4 + n^2$
- $j(n) = 3n^2 + 5n$

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

Svar:

- a. Alla funktioner ovan. Ingen funktion växer långsammare än något som är proportionerligt med n .
- b. $f(n)$, $h(n)$. Övriga funktioner växer långsammare än något som är proportionerligt med n^2 .
- c. $g(n)$, $i(n)$ funktioner växer långsammare än något som är proportionerligt med n^3 .

Fråga 2 (3pt)

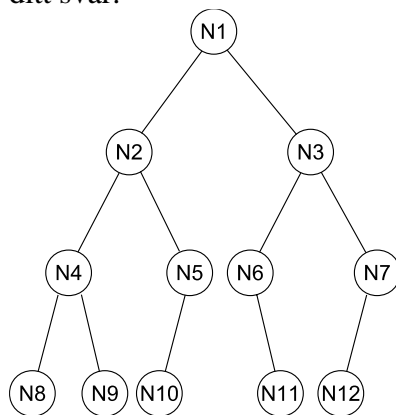
Hur många jämförelser behövs i värsta fall respektive bästa fall för hitta ett element i en sorterad lista med 16 element om du använder dig av binärsökning. Motivera kortfattat ditt svar.

Svar:

I bästa fall (om man har tur behövs bara en jämförelse). I värsta fall kommer det behövas $\log_2 16 + 1$ jämförelser. I binärsökning kan man i varje iteration bortse från hälften av återstående element. Efter $\log_2 16$ delningar på mitten har man endast 1 element kvar. Om man inte redan hittat elementet man söker efter behövs ytterligare 1 jämförelse.

Fråga 3 (2pt)

Vilken nod har minst respektive störst värde i följande binära sökträd. Motivera kortfattat ditt svar.



Svar: N8 är minst. N7 är störst. Man hittar minsta värdet genom att söka till vänster från roten tills man hittar en nod utan vänster barn. Man hittar noden med största värdet genom att söka till höger från roten tills man hittar en nod utan höger barn.

Fråga 4 (2p)

Stämmer det att en rutin (algoritm) för att lösa ett \mathcal{NP} -fullständigt problem kan användas som subrutin för att lösa vilket problem som helst i \mathcal{P} ? Motivera ditt svar. (2p)

Svar:

Ja, eftersom en rutin för att lösa ett \mathcal{NP} -fullständigt problem kan användas som subrutin för att lösa vilket problem som helst i \mathcal{NP} och \mathcal{P} är en delmängd av \mathcal{NP} .

Fråga 5 (3p)

Under föreläsningen om event-driven programmering fick ni uppgiften att skapa en finit tillståndsmaskin för en värme-regulator som har som uppgift att hålla temperaturen i en pool inom ett visst intervall. Dvs temperaturen skall aldrig vara högre än en viss temperatur och aldrig lägre än en viss temperatur.

Din uppgift här är att skapa en finit tillståndsmaskin för en värmeregulator som skall hålla temperaturen i ett rum inom ett visst intervall.

Du behöver presentera ett tillståndsdigram, samt beskriva de tillstånd, överföringar mellan tillstånd, samt villkor för överföring som du använt i din lösning.

Answer: Se föreläsningsanteckningar för pool-regulatorn. Det här är samma lösning.

Fråga 6 (4p)

EN: Look at the following C code, what will be printed after it is executed? Motivate why the output is the one you think is. (2pt)

SE: Titta på följande C-kod, vad kommer att skrivas ut efter att den har körts? Motivera varför utgången är den du tror är. (2 pkt)

```
#include <stdio.h>

int addOne(int var)
{
    var = var + 1;
}

int main()
{
    int a = 10;
    addOne(a);
    addOne(a);
    addOne(a);
    printf("%d ", a);
    return 0;
}
```

EN: We change the code a little, what would be printed now? Please motivate the answer. (2 pt)

SE: Vi ändrar koden lite, vad skulle skrivas ut nu? Motivera gärna svaret. (2 poäng)

```
#include <stdio.h>

int addOne(int* var)
{
    *var = *var + 1;
}

int main()
{
    int a = 10;
    addOne(&a);
    addOne(&a);
    addOne(&a);
    printf("%d ", a);
    return 0;
}
```

ANSWER:

- 1) The output is 10, this is because the function does change the value of a, because arguments are passed as value to functions in C, that is, their value is copied locally in the function, without modifying the variable outside the function.

- 2) In the second case the output is 13 because we pass the address of a, then the value inside the memory position is modified which survives after the execution of the function.

Fråga 6 (2p)

EN: Fill in the first column of this table placing the following different types of memory: DRAM, EEPROM, SRAM, PROM, NVRAM, Masked ROM, Flash, EPROM

SE: Fyll i den första kolumnen i denna tabell och placera följande olika typer av minne: DRAM, EEPROM, SRAM, PROM, NVRAM, Masked ROM, Flash, EPROM

Type	Volatile?	Writeable?	Cost (per Byte)	Speed
	Yes	Yes	Expensive	Fast
	Yes	Yes	Moderate	Moderate
	No	No	Inexpensive	Fast
	No	Once, with a device programmer	Moderate	Fast
	No	Yes, with a device programmer	Moderate	Fast
	No	Yes	Expensive	Fast to read, slow to erase/write
	No	Yes	Moderate	Fast to read, slow to erase/write
	No	Yes	Expensive	Fast

ANSWER:

The memory types are placed in this order: SRAM, DRAM, Masked ROM, PROM, EPROM, EEPROM, Flash, NVRAM

Fråga 7 (2p)

EN: Explain the difference between “Memory mapped I/O” and “Port mapped I/O”.

SE: Förklara skillnaden mellan "Minnesmappad I/O" och "Portmappad I/O".

ANSWER:

Memory mapped I/O is mapped into the same address space as program memory and/or user memory, and is accessed in the same way. Memory and IO share the same address and are used in the same way.

Port mapped I/O uses a separate, dedicated address space and is accessed via a dedicated set of microprocessor instructions. IO and memory use different address spaces, when accessing memory, the programmer must specify if it's memory or IO.

Fråga 9 (4pt)

You are developing the firmware for a microcontroller that controls the lights of a Christmas tree.



One of the pins of the microcontroller is connected to a button, which is used to control the device. The button is connected, on one side, to the microcontroller and on the other side to the voltage supply (V_{cc}).

A) (2 pt)

How should you configure your pin to detect when the button is pressed? You have the following 2 options to configure:

- Direction: can be *input* or *output*
- Connection: can be *direct*, *pulldown* or *pullup*.

B) (2 pt)

You want your lights to fade and blink smoothly, so you program the microcontroller to emit Pulse Width Modulation (PWM) pulses on a pin to control the light intensity. The PWM pulses are controlled by a timer, which, you estimate, should tick at 1KHz so that users would not notice any flickering.

Your microcontroller has a master clock of 8MHz, a prescale register of 8 bits and a compare register of 8 bits. How do you set the values for these two in order to obtain the desired frequency of 1KHz?

Answer:

A)

Input and pulldown. Pulldown is needed because when the button is not pressed, the pin is floating.

B)

Both the prescaler, P, and the compare register, C, accept values between 1 and 256.

Your target is $8M / (P \times C) = 1000$

Which means that $P \times C = 8000$

We can set, for example, $P = 200 \rightarrow C = 8000/200 = 40$