

F00 - Kursintroduktion, begrepp

5DV149 Datastrukturer och algoritmer

Niclas Börnin
niclas.borlin@cs.umu.se

Datavetenskap, Umeå Universitet

2020-01-22 Wed

Innehåll

- ▶ Kurspresentation:
 - ▶ Målsättning
 - ▶ Kursutvärdering
 - ▶ Upplägg
 - ▶ Kursöversikt
- ▶ Praktikaliteter
- ▶ Viktiga begrepp

Mina målsättningar

- ▶ Ni ska höja er ett par snäpp som problemlösare och programmerare!
- ▶ Ni ska förstå komplexitetsbegreppet:
 - ▶ "Om ett program tar 10s att sortera en lista på 1 miljon element, hur lång tid tar det för 2 miljoner?"
- ▶ Ni ska ha lärt er flera viktiga algoritmer och datatyper.
- ▶ Ni ska tycka att kursen är rolig!
- ▶ Eller: Senare kommer ni att se tillbaka på kursen och tycka att den var viktig.

Vad krävs av er för att nå dit? (1)

- ▶ Ge kursen en ärlig chans!
- ▶ Använd er tid och era pengar väl!
- ▶ Kom i tid till föreläsningarna!
- ▶ Läs boken!
 - ▶ En snabb genomläsning före och en noggrann efter ger minst stress.
- ▶ Se till att ni förstår vad som står i boken! Om något är oklart — fråga!

Vad krävs av er för att nå dit? (2)

- ▶ Gå på gruppövningarna!
 - ▶ Dom ger mer tid att jobba med typiska problem (och typiska tentauppgifter).
- ▶ Börja med laborationerna så fort som möjligt!
 - ▶ I planeringen framgår när all nödvändig teori än genomgången.
- ▶ Utnyttja handledningen! Vi har betalt för att hjälpa er.
- ▶ Säg till om det är något som inte funkar!
- ▶ Ni har ansvaret för att lära er! Vi finns här för att hjälpa er.

Tidsbudget

- ▶ Total tid: 200h, 10 veckor.
- ▶ Per vecka:
 - ▶ Total tid: 20h.
 - ▶ Föreläsningar: 0-4h (genomsnitt 3h/vecka).
 - ▶ Gruppövningar: 0-2h (1h/vecka).
 - ▶ Examination: 0-4h (0.6h/vecka).
 - ▶ **Egen tid: 12-20h (14h/vecka).** Att fördela mellan:
 - ▶Handledning: 2h.
 - ▶Läsa boken: (5 min/sida=3.15h/vecka).
 - ▶Egna övningar, OU.

Besättning

- ▶ Kursansvarig, föreläsare:
 - ▶ Niclas Börlin, niclas.borlin@cs.umu.se.
 - ▶ Kontor: MIT-huset, plan 2, C229.
 - ▶ Telefon: 090-786 68 32.
- ▶ Föreläsare
 - ▶ Anna Jonsson, aj@cs.umu.se
- ▶ Handledare, labrättare:
 - ▶ Hannah Devinney (English-speaking)
 - ▶ Fredrik Peteri
 - ▶ Ola Ringdahl
 - ▶ Lennart Steinvall

Cambrosajten

- ▶ <http://www.cambro.umu.se>
 - ▶ Schema.
 - ▶ Uppdateringar.
 - ▶ Formalia.
 - ▶ Gruppvöningar.
 - ▶ Föreläsningsmaterial.
 - ▶ Datatyper (kod) att använda under kursen.
 - ▶ Obligatoriska uppgifter.
 - ▶ Övningsquiz.
 - ▶ Forum
 - ▶Handledningsfrågor ska skrivas i forumet, inte mailas till handledarna.

Övningsquiz

- ▶ Kursmaterialet innehåller ett stort antal quiz som behandlar olika sektioner av materialet.
- ▶ Quizen har ingen tidsbegränsning och går att göra hur många gånger som helst.
- ▶ I allmänhet mycket uppskattade.
- ▶ Som extra morot så är lösningsförslag till gruppövningar m.m. låsta tills ni gjort vissa quiz.
- ▶ Quizen gör att ni "tvingas" komma i gång med materialet tidigt under kursen.
 - ▶ Erfarenheten är att det är bra för inläringen.

Praktikaliteter

- ▶ Akademisk kvart för starttider
 - kl. 8 betyder 8.15
 - kl. 10.00 betyder 10.00
- ▶ Registrera er på kursen!
 - ▶ Icke registrerade studenter kommer att tas bort inom en vecka.
- ▶ Lär er er cs-användare!
 - ▶ All mail-kommunikation med oss behöver den informationen.
 - ▶ Hantering av mail utan cs-användare kan bli fördröjd eller inte alls.

0. Har du gått kursen förut?

- ▶ Ja, och jag vill skriva **tentan** i år:
 1. Omregistrera dig på kursen.
 2. Invänta att anmälningfönstret för tentan öppnas.
 - ▶ Anmälningfönstret öppnar 20 (35?) dagar före tentamensdatum och stänger 10 dagar före.
 3. Anmäl dig till tentan.
- ▶ Ja, och jag vill göra **labbarna** i år:
 1. Omregistrera dig på kursen.
 2. Gör quiz "0 - Tillgodoräknande av gamla godkända labbar" (om du har gamla godkända labbar).
 3. Invänta att labspecifikationerna publiceras.
 4. Gör labbarna och lämna in dem enligt **årets** specifikation.

1. Omregistrering

- ▶ För att omregistrera dig på kursen, maila nedanstående information till studentexp@cs.umu.se:
 - ▶ Ditt **personnummer**.
 - ▶ Ditt **UmU-id**¹, t.ex. kasv0042.
 - ▶ Kurskod: **5DV149**.
 - ▶ Anmälningsskod för detta kurstillfälle: **57314vt19**.
 - ▶ **Anmälningsskod** för kurstillfället **när du först gick kursen**, t.ex. 57314vt18.
 - ▶ Om du inte kommer ihåg anmälningsskoden, skriv år och läsperiod alt. startmånad för kursen i stället, t.ex. 2018, LP3 eller januari 2018.
- ▶ Ange **Omregistrering 5DV149** som ärende (*subject*).
- ▶ En administratör kommer att omregistrera er i Ladok och lägga till er på kursens Cambro-sida.

¹<http://www.it.umu.se/vara-tjanster/umu-id/>

2. Ansökan om tillgodoräknande av labresultat

- ▶ Om du har gått kursen tidigare utan att få **samtliga** laborationer godkända så måste du göra **årets** laborationer.
- ▶ Men... om årets laborationer liknar de från ditt tidigare tillfälle så *kan* du få tillgodoräkna dig godkända resultat.
- ▶ Gör quiz "Tillgodoräknande av gamla godkända labbar".
- ▶ Kursansvarig verifierar godkända resultat och markerar dem som tillgodoräknade.
- ▶ **OBS!** Gör detta så snart som möjligt! Sen ansökan kan medföra att ni missar ett inlämningstillfälle!

Examination (1) — salstentamen

- ▶ Tre tillfällen i år:
 1. Under kursens gång. Tidigarelagd för att ge mer utrymme till OU5.
 2. Senare under våren.
 3. Uppsamling i augusti.
- ▶ Ytterligare tillfällen under kursen nästa år.

Examination (2) — labbar

- ▶ Fem stycken obligatoriska uppgifter (labbar):
 - ▶ OU1: Komplexitetsanalys – hur argumenterar ni att er algoritm är bättre än grannens?
 - ▶ OU2: Testning – hur ni slår ihjäl någon annans kod
 - ▶ OU3: Debugging – hur ni slår tillbaka
 - ▶ OU4: Tabeller – hur ni byter implementation på en datatyp utan att nån märker det.
 - ▶ OU5: Grafer – hur hittar google maps till Skövde?
- ▶ OU1 redovisas som quiz i Cambro.
- ▶ OU3 redovisas muntligt. Publiceras samtidigt som OU2 för att ge er chansen att använda debugger på OU2.
- ▶ OU2, OU4, OU5 redovisas via inlämning i labres.

Labbar (1)

- ▶ Labspecifikationen publiceras senast så fort vi har gått igenom den teori ni behöver.
- ▶ Schemat, handledningstillfällena och deadlines är planerade utifrån att ni **börjar med labben på en gång.**
- ▶ För varje lab har ni **tre chanser i år:**
 - ▶ Första inlämning.
 - ▶ Planerad till 10–21 dagar efter publicering.
 - ▶ Andra inlämning.
 - ▶ Typiskt 14 dagar efter första inlämning för resp. lab.
 - ▶ Vi behöver tid att bedöma ("rätta") labbarna.
 - ▶ Ni har rätt till 5 arbetsdagar att komplettera.
 - ▶ Tredje inlämning.
 - ▶ Ett tillfälle i augusti. Samma för alla labbarna. **Inga kompletteringsmöjligheter efter detta.**
- ▶ Fler chanser **nästa år** då kursen ges.

Labbar (2)

- ▶ Grov uppskattning av tidsfördelning mellan labbarna:
 - ▶ OU1: 10h.
 - ▶ OU2: 18h.
 - ▶ OU3: 10h.
 - ▶ OU4: 31h.
 - ▶ OU5: 48h.
- ▶ Följande lab är reserverade för er måndagar och torsdagar 13-16
 - ▶ MA416, MA426, MA446 ².

²Undantag: torsdag 2020-02-13 är MA446 upptagen 15-16.

- ▶ "Datatyper och algoritmer", Janlert, Wiberg, 2000.



Föreläsningar

- ▶ Några samläses med python-kursen.
- ▶ Vi kommer att sträva efter att lägga upp föreläsningsanteckningar dagen före respektive föreläsning.
 - ▶ Två versioner
 - ▶ En "handout" med fyra bilder/sida. För utskrift.
 - ▶ En "slides" med en bild per sida + animeringar. **Skriv inte ut!**
 - ▶ Fjolårets föreläsningsanteckningar finns i en undermapp.
- ▶ Ni vinner **mycket** på att skumma igenom boken och föreläsningsantecknarna före föreläsningen!

Gruppövningar ("lektioner" i schemat)

- ▶ Alla gruppövningar ger er chans att öva på och ställa frågor kring typiska problem (och tentauppgifter).
- ▶ Innehåll:
 - Gruppövning 1 Minnesallokering och filer.
 - Gruppövning 2 Komplexitetsanalys.
 - Gruppövning 3 Algoritmkonstruktion.
 - Gruppövning 4 Träd.
 - Gruppövning 5 Gamla tentafrågor.

Handledning

- ▶ Vi kommer att bemanna labben normalt kl. 13-15 på måndagar och torsdagar med tre handledare.
- ▶ Vid utökat behov, handledning 13-16 och en handledare till.
- ▶ Handledning endast i labben enligt ovan.
- ▶ Ingen handledning utanför handledningstid.
 - ▶ Använd forumet!

Kursutvärdering från vt-19 (1)

- ▶ Positivt, bör behållas:
 - ▶ Föreläsningsanteckningarna.
 - ▶ Kursboken.
 - ▶ Övningsquizen!
 - ▶Handledningen
 - ▶ Obligatoriska uppgifter, speciellt OU5.
 - ▶ Att få jobba i par på OU5.
 - ▶ Läraren.
 - ▶ Tidigarelagd tenta.
 - ▶ *Lärarens engagemang och sätt att bidra med lite komedi, det gör det mindre spännigt i klassen och man blir nyfiken på att lyssna.*

Kursutvärdering från vt-19 (2)

- ▶ Vad kan förbättras? Ge gärna tips på hur!
 - ▶ Tentan för omfattande.
 - ▶ Bättre med senare tenta.
 - ▶ Mindre omfattande OU5.
 - ▶ Fler handledningstimmar inför labbarna, speciellt OU5.
 - ▶ *Lagt till fler timmar inför deadline.*
 - ▶ *Mer handledningstid. Lätt att fastna med uppgifter flera dagar då det inte fanns någon handledning att tillgå tis-ons-halva tors.*
 - ▶ *Lagt till ett forum. Använd det!*
 - ▶ Lång kö till handledare.
 - ▶ För stort fokus på rapportskrivande på OU4 och OU5.
 - ▶ *OU4 i par.*

Kursutvärdering från vt-19 (3)

- ▶ Vad kan förbättras? Ge gärna tips på hur!
 - ▶ *Ge ut facit till gruppuppgifterna utan att behöva klara av quiz.*
 - ▶ *Nope!*
 - ▶ För stort hopp från C-programmeringen.
 - ▶ *Trycka mer på att kodexempel finns att ladda ner.*

Kursutvärdering från vt-19 (4)

- ▶ Övriga kommentarer:
 - ▶ *Det här den bästa kursen hittills på min utbildning. Även om DOAn krävde många timmar av slit, förvirring, och energi så känner jag att jag har en god grund att stå i inför kommande datavetenskapskurser.*

Hur får jag svar på mina frågor?!!

1. Kolla materialet! Cambro, kursboken, ...
2. Skriv i forumet!
3. Sök! Wikipedia, google, ...
4. Fråga dina kursare! (kopiera dock inte deras kod)
5. Fråga vid nästa föreläsning, ev. i pausen! Skicka gärna ett mail till mig före nästa föreläsning. (Räkna dock inte med svar innan.)
6. Fråga handledarna på handledningstid!
 - ▶ Om ni står i kö, lyssna på “aktiv” handledning – den kanske handlar om ert problem!
 - ▶ Leta i “kön” efter någon annan med samma fråga.
 - ▶ Vi ska försöka organisera så bra som möjligt.

Tentamen, föreläsningsanteckningarna och boken

- ▶ Ni tenteras på FSR (förväntade studieresultat) enligt kursplanen!
 - ▶ ...ej på föreläsningsanteckningarna...
 - ▶ Däremot är föreläsningsanteckningarna vår (min) bästa sammanfattning av kursmaterialet (kursboken, m.m.).
 - ▶ ...inte ens på boken...
 - ▶ Däremot är boken den bästa sammanfattning av kursinnehållet vi vet.
- ▶ Väljer ni att plugga föreläsningsanteckningarna utan att läsa i boken så tar ni en risk.

Fusk och plagiat

- ▶ Från tidigare kursvärdering:
 - ▶ *Tycker att det blir konstig stämning och då man inte får prata med varandra om uppgifterna så blir det svårt att veta om man följer reglerna eller inte.*
- ▶ Reglerna säger att uppgifterna ska utföras *enskilt* (eller i par):
 - ▶ Ni ska förstå lösningen.
 - ▶ Ni ska kunna motivera lösningen.
 - ▶ Ni ska ha skrivit koden själv.
 - ▶ Ni ska ha skrivit rapporten själv.
 - ▶ Ni får diskutera algoritmer med varandra.
 - ▶ Ni får diskutera kodlösningar med varandra, men skriv inte av varandras kod.

Kursinnehåll

Datastrukturer och Algoritmer

- ▶ Vi ska ge er möjlighet att:
 - ▶ strukturera hur ni tänker kring programmering.
 - ▶ se strukturer, hitta likheter för att kunna bygga större och mer komplexa program.
 - ▶ lära er nya byggblock och lära er i vilka situationer de passar.

Innehållsöversikt

- ▶ Algoritmer:
 - ▶ Pseudokod som ett sätt att beskriva algoritmer.
 - ▶ Tids- och rumskomplexitet.
 - ▶ Komplexitetsanalys (experimentell och asymptotisk).
 - ▶ Designprinciper.
 - ▶ Fokus: Sökning och sortering
 - ▶ Mål: Ni ska kunna välja och implementera en bra algoritm som passar för en given situation.
 - ▶ Abstrakta datatyper:
 - ▶ Lista, kö, prioritetsskö, stack, träd, graf, heap, tabell, sträng, mängd, sökträd, tries, ...
 - ▶ ... med tillhörande algoritmer, begrepp, användningsområden, relationer, beskrivningssätt m.m.
 - ▶ Testning och felsökning.

Att konstruera program (1)

- ▶ Att bygga program \approx bygga broar.
- ▶ Ett brobygge behöver:
 - ▶ Specifikation av målet.
 - ▶ Beskrivning av lösningen.
 - ▶ Material att utgå ifrån.
 - ▶ Verktyg och tekniker för att bearbeta materialet.
 - ▶ Uppföljning av den färdiga lösningen.

Att konstruera program (2)

- ▶ Problembeskrivning.
- ▶ Systemdesign.
- ▶ Modelleringsverktyg:
 - ▶ Datatyper — representerar datat i programmet.
 - ▶ Algoritmer — en stegvis plan för att utföra något, en modell av hur man löser problemet.

Vad är då en datatyp?

- ▶ Objekt *och* operationer.
 - ▶ Vilka objekt?
 - ▶ Vad vill vi modellera/representera/abstrahera?
 - ▶ Vad vill vi göra med dem?
 - ▶ Vilka operationer/metoder?
- ▶ Exempel:
 - ▶ Heltal med funktioner (+, -, etc.), relationer ($>$, $<$, $=$, \leq , \geq , etc.), predikat (udda, jämn, positiv, negativ, etc.).
 - ▶ Mängd och lägg till, slå upp, snitt, etc.
 - ▶ Kö och stoppa in (sist), ta ut (först), etc.

Gränssnitt (*interface*), definition

- ▶ Vad är ett gränssnitt?
 - ▶ Kontaktyta.
 - ▶ Gränsen mellan två eller flera delar.
 - ▶ Överenskommelse – oberoende av vem, var och hur ska det “passa”.
 - ▶ Jämför reglagen i en bil.

Gränssnitt, egenskaper

- ▶ Ett gränssnitt separerar
 - ▶ Funktion och implementation.
 - ▶ Användande och skapande.
 - ▶ Specifikation och konstruktion.
- ▶ Inom datavetenskap:
 - ▶ Mellan centralenheter och periferenheter. Ex. USB-sladd.
 - ▶ Mellan människan och maskinen. Ex. tangentbord, skärm.
 - ▶ På denna kurs: mellan mjukvarukomponenter:
 - ▶ Funktioner/metoder.
 - ▶ Datatyper.

Begrepp (1)

- ▶ Data – objekt som bär information. Ex. sträng som representerar namnet på ett hotell, heltal som representerar kraften som genereras av en raketmotor.
- ▶ Datatyp – “värdemängd” samt operationer.
- ▶ Sammansatt datatyp vs. enkel datatyp.
 - ▶ En sammansatt datatyp har delar (element), strukturerade på visst sätt.
 - ▶ Ex. kö av filer (printerjobb), struktur som beskriver ett hotell.
 - ▶ Homogen (alla element har samma typ) eller heterogen (olika typ).
- ▶ Ordnad/oordnad datatyp
 - ▶ I en ordnad datatyp har elementen en inbördes ordning, t.ex. en lista.
 - ▶ I en oordnad datatyp saknar elementen en inbördes ordning, t.ex. en mängd.

Begrepp (2)

- ▶ Ordnad/sorterad datatyp
 - ▶ I en sorterad datatype bestäms ordningen på elementen av dess *värden*.
- ▶ Abstrakta datatyper – en datatyp definierad helt av dess gränssnitt. Behöver inte ens vara implementerad. Ex. prioritetskö.
- ▶ Konkret datatyp – implementation av en abstrakt datatyp. Ex. en kö implementerad som länkad lista.
- ▶ Konstruerad datatyp – ex. schackpjäs implementerad som ett heltal eller ett hotell implementerad som en `!struct!` med namn, m.m.
- ▶ Fysisk — implementerad i språket/hårdvaran, ex. 8-bitars heltal, 64-bitars flyttal, pekare.
- ▶ Implementerad datatyp – färdig att använda.

Övning

- ▶ Ge exempel på fler sammansatta datatyper
 - ▶ Homogena
 - ▶ Heterogena
- ▶ Är en sorterad datatyp alltid ordnad? Tvärtom?
- ▶ Vad är skillnaden mellan en konkret datatyp och en implementerad datatyp?
- ▶ Vad är skillnaden mellan en fysisk datatyp och en implementerad datatyp?