

Kurs-PM hösten 2019 för DD2350 Algoritmer, datastrukturer och komplexitet, adk19

Denna kurs ger en introduktion till *teoretisk datalogi* som är ett starkt forskningsområde på KTH. Du kommer att stöta på några av våra forskningsresultat i kursen.

Du får lära dig mer om algoritmkonstruktion och får se några ganska komplicerade, men mycket användbara, algoritmer. Komplexitetsdelen av kursen handlar om hur man undersöker vilka problem som kan lösas (i rimlig tid) med datorns hjälp, vilka som tar orimligt lång tid och vilka som inte kan lösas med en dator över huvud taget.

Problem som är för svåra för att lösa exakt kan ibland lösas approximativt. Du kommer att få se exempel på några approximationsalgoritmer och några problem som är så svåra att dom inte ens kan approximeras i rimlig tid.

Innehåll och lärandemål

Kursinnehåll *

Konstruktionsprinciper för algoritmer: Dekomposition, giriga algoritmer, dynamisk programmering, lokal och total sökning. Algoritmanalys. Approximationsalgoritmer och heuristiker. Tillämpningar med algoritmer för problem på mängder, grafer, aritmetik och geometri. Implementation av algoritmer.

Datastrukturer: Repetition av hashtabeller och heapar; balanserade träd, bloomfilter. Användning och implementation av datastrukturer. Beräkningsbarhet och komplexitet: Reduktionsbegreppet, komplexitetsklasserna P (polynomisk tid) och NP (ickedeterministisk polynomisk tid). NP-fullständiga problem, oavgörbara problem. Hur man kan hantera problem med hög komplexitet. Ämnesterminologin på svenska och engelska.

Lärandemål *

Efter godkänd kurs ska studenten kunna

- utveckla och implementera algoritmer med datastrukturer och analysera dem med avseende på korrekthet och effektivitet,
- jämföra alternativa algoritmer och datastrukturer med hänsyn till effektivitet och pålitlighet,

- definiera och översätta centrala begrepp som P, NP, NP-fullständighet och oavgörbarhet,
- jämföra problem med hänsyn till komplexitet med hjälp av reduktioner,
- hantera problem med hög komplexitet

i syfte att

- självständigt kunna konstruera datorprogram som effektivt utnyttjar tid och minne,
- i yrkeslivet kunna identifiera och angripa problem som är orealistiskt resurskrävande eller inte alls går att lösa med dator.

Kursens pedagogiska upplägg

- Studera på det sätt som är effektivast för dig! Allt föreläsnings- och övningsmaterial finns tillgängligt i förväg.
- Koncentrerade entimmesföreläsningar med läsanvisningar. Kom förberedd och var vaken för bästa resultat! I avsnittet dynamisk programmering används omvänd undervisning (flipped classroom).
- Övningsuppgifter med fullständiga lösningar. Övningsgrupper med svårighetsgradering. Ett urval av uppgifterna löses på övningarna, resten lämnas för egen övning.
- Momenten i kursen tränar verkliga arbetssituationer för bättre autenticitet.
- Aktiverande färgfrågor på föreläsningarna och kontinuerlig examination med labbteoriuppgiftsredovisning inför varje datorlabb gör att du automatiskt hänger med i kursen.
- Undervisning byggd på pedagogisk forskning - en hel doktorsavhandling om ADK las fram 2014!
- Mårelaterade betygskriterier; välj själv betyg!
- Gott om tid för labbar och mästarprov, ingen stressad tentasituation.

Kursen består av 33 föreläsningar och 14 övningar. Alla föreläsningar efter första veckans tre föreläsningar är egentligen entimmesföreläsningar (men av schematekniska skäl har vid ett tillfälle två entimmesföreläsningar kommit att hamna direkt efter varandra). Viggo Kann [VK] och Stefan Nilsson [SN] delar på föreläsningarna. Följande tabell visar vad som kommer att behandlas under föreläsningarna och övningarna. För varje föreläsning anges vilket material i kurslitteraturen som behandlas. Du bör ha skummat det innan du kommer till föreläsningen för att ha riktig glädje av föreläsningen. Föreläsning 9-11 om dynamisk programmering har omvänd undervisning, så videor ska ses och uppgifter ska göras före varje föreläsning

Kopplingar till examensmål

ADK-kursen bidrar till att följande mål för civilingenjörsexamen uppfylls:

- visa kunskap om det valda teknikområdets (datatekniks) vetenskapliga grund och beprövade erfarenhet

(Kursens område utgör själva kärnan i datalogins vetenskapliga grund. Både teoretiskt sett optimala algoritmer och datastrukturer och erfarenhetsmässigt goda sådana behandlas i kursen.)

- visa brett kunnande inom det valda teknikområdet (datateknik), inbegripet kunskaper i matematik och naturvetenskap, som väsentligt fördjupade kunskaper inom vissa delar av området

(Kursen är på avancerad nivå och bygger vidare på flera tidigare kurser i matematik och datalogi. Den ger fördjupade kunskaper inom teoretisk datalogi och algoritmkonstruktion.)

- självständigt identifiera, formulera och hantera komplexa frågeställningar (Matematisk modellering och konstruktion av heuristiker för hantering av svåra problem examineras i kursen.)
- skapa, analysera och kritiskt utvärdera olika tekniska lösningar (Analys av komplexitet och korrekthet för algoritmer är centralt i kursen.)
- planera och med adekvata metoder genomföra kvalificerade uppgifter inom givna ramar (Individuell självständighet i kreativ problemlösning - expertsituationen - visas i mästarproven. Den teoretiska grunden och verktyg för bedömning av problems komplexitet ges och examineras.)
- muntligt och skriftligt klart redogöra för och diskutera sina slutsatser och den kunskap och de argument som ligger till grund för dessa (Labbar redovisas med programkod och muntligt i par. Mästarproven redovisas med skriftlig rapport och muntligt individuellt. Förmågan att förklara och genomföra matematiska resonemang, vilket är ett explicit program mål för datateknikprogrammet, examineras vid mästarproven. Ämnesterminologin på både svenska och engelska examineras.)
- visa insikt i teknikens möjligheter och begränsningar (I kursen behandlas NP-svåra och oavgörbara problem och studenterna övar på att visa att problem är svåra och att finna metoder att ändå hantera dessa problem med dagens teknik.)

Undervisningsspråk

Svenska. Kursböckerna är på engelska. Använd listan med nyckelbegrepp nedan för att se vilka svenska och engelska termer som motsvarar varandra.

Detaljschema och läsanvisningar

- **KT**=Kleinberg-Tardos, oavsett om det är KOrig eller KTnie
- **KOrig**=Kleinberg-Tardos International Edition (2006) eller motsvarande amerikanska utgåva
- **KTnie**=Kleinberg-Tardos New International Edition (2014), när denna skiljer sig från den tidigare utgåvan (kapitel 5 *Divide and Conquer* kommer före kapitel 4 *Greedy Algorithms* och kapitel 13 *Randomized Algorithms* kommer före kapitel 12 *Local Search*)
- **Sup**=supplementet Algorithms and Complexity

Period 1

- F1 ([timme 1](#), [timme 2](#)) 27 augusti (2 timmar)
[SN+VK] Introduktion till kursen. Repetition av algoritmanalys, beräkningsmodeller, bitkostnad, enhetskostnad. (KT: 29-56)
Mer material: [tidskomplexitet](#), [bitkostnad](#), [ordonotation](#), [mästarsatsen](#).
- F2 28 augusti (2 timmar)
[SN] Repetition av sortering ([animering 1](#), [animering 2](#)). (KOrig: 209-221/KTnie: 115-127)
[Effektiv kodning](#) och testning. Labbpartnermatchning i pausen.
Mer material: [insättningsortering](#), [quicksort](#). Se även [Diverse länkar](#).
- F3 29 augusti (2 timmar)
[VK] Datastrukturer: [repetition](#), [hashning](#), praktiska datastrukturer, [trie](#) ([animering](#))
(KT: 57-65) Latmanshashning, [skipplistor](#). ([Sup: 77-83](#)) Labbpartnermatchning i pausen.
- Ö1 30 augusti
Algoritmanalys.
- F4 2 september
[SN] Grafer: [djupetförstsökning](#), [breddenförstsökning](#). (KT: 73-107)
Mer material: [grafteori och grafsökning](#), [grafpaket i Go](#)
- F5 3 september
[VK] Datastrukturer: bloomfilter. Tillämpning: [rättstavning](#).
Mer material: [bloomfilter](#)
- F6 4 september
[SN] Korrekthetsbevis.
Mer material: [invarianter](#), [induktion](#).
- Ö2 4 september
Datastrukturer och grafer. **Teoriredovisning för labb 1.**
- F7 10 september
[SN] Algoritmkonstruktion: giriga algoritmer, totalsökning. (Sup: 31-48, KOrig: 115-136, 183-188/KTnie: 157-179, 225-230)
- F8 11 september
[SN] Algoritmkonstruktion: dekomposition. (KOrig: 221-234, 242-246/KTnie: 127-140, 148-152)
- F9 12 september (OBS! [förberedelse krävs](#))
[SN] Algoritmkonstruktion: dynamisk programmering, del 1. (KT: 251-260)
Visualiseringar: [Fibonaccitalen](#).
- Ö3 12 september
Dekomposition och dynamisk programmering.
- Labb 1 13 september
Konkordans, redovisning.

[. . .]

Nyckelbegrepp

Här är 50 facktermer i ADK-kursen som inte har en uppenbar direktöversättning.

algoritm	algorithm
approximationskvot	approximation ratio
approximerbarhet	approximability
beräkningsbarhet	computability
beräkningsmodell	computational model
beslutsproblem	decision problem
bitkostnad	bit cost
datastruktur	data structure
dekomposition	divide and conquer
delmängdssumma	subset sum
dynamisk programmering	dynamic programming
enhetskostnad	unit cost
förberäknad (funktion)	precomputed (function)
girig algoritm	greedy algorithm
grafgenomgång	graph traversal
grannlista	adjacency list
grannmatris	adjacency matrix
heuristik	heuristics
hörn	vertex
ickedeterministisk	non-deterministic
kant	edge
kantmatris	incidence matrix
komplexitet	complexity
konjunktiv normalform	conjunctive normal form
konstruktionsproblem	construction problem
källa	source
latmanshashning	lazy hashing (<i>ingen standardterm</i>)
målfunktion	objective function
mängdpartitionering	partition problem
mästarsatsen	Master theorem
NP-fullständig	NP-complete
oavgörbar	undecidable
oberoende mängd	independent set
optimeringsproblem	optimization problem

polynomisk reduktion	polynomial reduction
polynomisk tid	polynomial time
prioritetskö	priority queue
probleminstans	problem instance
rekursivt uppräknelig	recursively enumerable
restkapacitet	residual capacity
rimlig tid	feasible time
räknesortering	counting sort
satisfierbar	satisfiable
simulerad härdning	simulated annealing
slumpeliminering	derandomization
spännande träd	spanning tree
totalsökning	exhaustive search
tuff motståndare	adversary
undre gräns	lower bound
utlopp	sink
verifierbar	verifiable
övre gräns	upper bound

Kurslitteratur och förberedelser

Särskild behörighet *

Programmering och datalogi motsvarande
DD1338/DD1320/DD1321/DD1325/DD1327/DD1339/ID1020.

Rekommenderade förkunskaper

För vissa av kursens labbar måste ett snabbare programspråk än Python användas, varför kunskaper i Java eller C/C++ rekommenderas. Diskret matematik motsvarande SF1671 och SF1688 eller någon av SF1630, SF1662, SF1679 (kan läsas parallellt). Sannolikhetsteori och statistik motsvarande SF1901. Logik motsvarande DD1350/DD1351 rekommenderas men är inte nödvändigt.

Kurslitteratur

- *Algorithm Design* av Kleinberg-Tardos, Pearson, 2014, ISBN 978-1292023946.

- Det specialtryckta supplementet *Algorithms and Complexity, a supplement to Algorithm Design*, Pearson Custom Publishing, ISBN 978-1847764126. Säljs bara på kårbokhandeln.

Funktionsnedsättning

Om du har en funktionsnedsättning kan du få stöd via Funka:

<https://www.kth.se/student/studentliv/funktionsnedsattning>

Informera dessutom kursledaren om du har särskilda behov. Visa då upp intyg från Funka.

Examination och slutförande

Betygsskala *

A, B, C, D, E, FX, F

Examination *

- LAB1 - Laborationsuppgifter, 4,0 hp, betygsskala: A, B, C, D, E, FX, F
- MAS1 - Individuellt mästarpöv, 1,5 hp, betygsskala: A, B, C, D, E, FX, F
- MAS2 - Individuellt mästarpöv, 1,5 hp, betygsskala: A, B, C, D, E, FX, F
- TEN1 - Teoritentamen, 2,5 hp, betygsskala: P, F

Examinator beslutar, baserat på rekommendation från KTH:s samordnare för funktionsnedsättning, om eventuell anpassad examination för studenter med dokumenterad, varaktig funktionsnedsättning.

Examinator får medge annan examinationsform vid omexamination av enstaka studenter.

Examinator

[Viggo Kann](#)

Etiskt förhållningssätt *

- Vid grupparbete har alla i gruppen ansvar för gruppens arbete.
- Vid examination ska varje student ärligt redovisa hjälp som erhållits och källor som använts.
- Vid muntlig examination ska varje student kunna redogöra för hela uppgiften och hela lösningen.

I denna kurs tillämpas EECS hederskodex, se:

<http://www.kth.se/eecs/utbildning/hederskodex>

Målrelaterade betygskriterier

mål	E	D	C	B	A
<i>utveckla algoritmer med data-strukturer</i>	för enkla problem givet en konstruktionsmetod	för icke-triviala problem givet ledtråd	för icke-triviala problem	[A-kriteriet] givet ledtråd	för svårare problem med den metod som passar bäst
examineras med labbar (för nivå E), mästarpöv 1 och muntlig tenta					
<i>implementera algoritmer med data-strukturer</i>	efter funktions-specifikation och efter detaljerad algoritmisk specifikation, med hänsyn taget till effektivitet	med vissa krav på leveranstid	med höga krav på leveranstid		
examineras med labbar					
<i>analysera algoritmer med avseende på effektivitet</i>	förklara principerna, analysera enklare algoritmer	analysera svårare algoritmer givet ledtråd	analysera svårare algoritmer		
examineras med labbar och teoritenta (för nivå E), mästarpöv 1 och muntlig tenta					
<i>analysera algoritmer med avseende på korrekthet</i>	förklara principerna, förklara ett givet korrekthetsbevis	framställa grundläggande idé för korrekthetsbevis	framställa grundläggande idé och givet ledtråd genomföra fullständiga korrekthetsbevis	[A-kriteriet] givet ledtråd	genomföra fullständiga korrekthetsbevis med invarianter
examineras med mästarpöv och muntlig tenta					
<i>jämföra alternativa algoritmer och datastrukturer med hänsyn till effektivitet och pålitlighet</i>					
examineras med labbar, teoritenta och mästarpöv 1					
<i>definiera och översätta centrala begrepp som P, NP, NP-fullständighet och oavgörbarhet</i>					
examineras med teoritenta och mästarpöv 2					

<i>jämföra problem med hänsyn till komplexitet med hjälp av reduktioner</i>	förklara principerna, utföra enklare reduktioner mellan givna problem	[C-kriteriet] givet ledtråd	visa NP-fullständighet eller oavgörbarhet	[A-kriteriet] givet ledtråd	utföra konstruktionsreduktioner
examineras med teoritentor och labb 4 (för nivå E), mästarprov 2 och muntlig tenta					
<i>förklara hur man kan hantera problem med hög komplexitet</i>	förklara principerna, konstruera enkla heuristiker		konstruera och analysera heuristiker		konstruera och analysera mer avancerade heuristiker
examineras med teoritentor och labb 5 (för nivå E) och betygshöjande del av labb 5 (för A och B)					

Kursen har tre graderade provmoment: MAS1 (individuellt mästarprov 1), MAS2 (individuellt mästarprov 2) och LAB1 (datorlabbar). TEN1 (teoritentan) har inte graderat betyg.

Slutbetyget är medelbetyget av de tre graderade betygen, avrundat till närmaste betyg.


Examinationsdetaljer

Laborationer

Fem obligatoriska datorlabbar ingår i kursen. Dessa utgör momentet LAB1. Labbarna ska göras i tvåpersonsgrupper, men enpersonsgrupper kan godkännas av kursledaren i undantagsfall. Den betygshöjande delen på labb 5 måste dock göras individuellt. Varje labb som redovisas och godkänns senast det labbtillfälle som finns angivet på labben ger en så kallad labbleveranspoäng. Den som fått 4-5 labbleveranspoäng (dvs har levererat minst fyra av labbarna i tid) får betyg C på momentet LAB1. Den som får 2-3 labbleveranspoäng får betyg D. Betyg C kan höjas till betyg A eller B med den betygshöjande extralabben som är en påbyggnad på labb 5, som ska göras och redovisas individuellt vid ett speciellt labbredovisningstillfälle i januari.

På varje labb finns dessutom ett antal frivilliga teoriuppgifter. Teoriuppgifterna redovisas skriftligt och muntligt på övningstillfällena (ingen annan möjlighet till redovisning ges) och ger en teori-poäng var, som ger bonus på tentorna under närmaste läsåret.

Det finns schemalagda labbtillfällen under hela kursen. Det kommer att finnas handledare tillgängliga på dessa labbpass. Börja att göra labbarna i god tid och fråga handledarna om du får problem. Du kan i princip redovisa alla labbarna vid alla labbtillfällen, men under det sista labbtillfället för varje labb, som är fyra timmar långt, prioriteras redovisningar av den labben.

I Canvasmodulen labbar ligger labblydelserna för kursen. Där finns också ett [labbkvitto](#)  (där labbhandledaren kan signera att du är godkänd på labbar) som du ska ta med vid varje redovisning.

Individuella uppgifter: mästarprov

Två obligatoriska individuella uppgifter, *mästarprov*, kommer att ges. Dessa ska lösas *individuell*t och redovisas både skriftligt och muntligt. Skriftliga lösningar till dessa uppgifter ska lämnas in i Canvas senast den tid som anges på uppgiftslydelsen. Den muntliga redovisningen kommer att ske några dagar senare för någon av assistenterna på en tid som ska bokas i förväg i Canvas.

Varje mästarprov består av tre uppgifter av olika svårighetsgrad som testar betygsriterierna för E, C respektive A.

Inför varje mästarprov ges ett frivilligt övningsmästarprov som kan lösas i grupp och som redovisas vid speciella mästarprovövningar, se detaljschemat. Godkänd redovisning av ett övningsmästarprov ger en teoripoäng. Totalt kan alltså två teoripoäng fås från mästarprovövningar.

Den som inte godkänts på ett mästarprov får möjlighet att göra ett nytt i slutet av kursen, men kan då bara få betyg E på mästarprovet. Dessa ommästarprov läggs upp i Canvas i samband med ordinarie teoritentan och ska redovisas både skriftligt och muntligt i omtentaveckan i januari.

Du kan se dina resultat på redovisade uppgifter i kursen under *Omdömen* i Canvas.

Teoritentan

Ordinarietentan går den 17 december 2019 klockan 9.00 i sal F1 och F2. Första omtentatillfälle är i påskperioden.

Tentan (momentet TEN1) är en teoritentan utan hjälpmedel. Tentans uppgifter är alla på E-nivå, dvs det går inte att få mer än godkänt på tentan. För godkänt krävs minst 13 av 14 poäng inklusive poäng på uppgifter som visar uppfyllelse av lärandemålet *definiera och översätta centrala begrepp*. Den som får 11 eller 12 poäng eller som får minst 13 poäng men har missat måluppfyllelsen får möjlighet att komplettera till godkänt. Teoripoängen som samlats genom labbteoriredovisningar (upp till 5 teoripoäng) och övningsmästarprov (upp till 2 teoripoäng) läggs till poängen på teoritentan på alla tentor inom ett år från kursstart.

Teoritentans uppgifter testar följande betygskriterier på nivå E:

- *analysera algoritmer med avseende på effektivitet*: förklara principerna, analysera enklare algoritmer
- *jämföra alternativa algoritmer och datastrukturer med hänsyn till effektivitet och pålitlighet*
- *definiera och översätta centrala begrepp som P, NP, NP-fullständighet och oavgörbarhet*
- *jämföra problem med hänsyn till komplexitet med hjälp av reduktioner*: förklara principerna
- *förklara hur man kan hantera problem med hög komplexitet*: förklara principerna

Vi rekommenderar alla att titta på E-delen [senaste årens extentor](#) för att bättre förstå hur uppgifterna kan se ut. Lösningförslag finns bara till ordinarie tentorna.

Skrivtiden är 90 minuter. Direkt efter tentan vidtar obligatorisk genomgång av lösningarna till tentan och kamraträttning. Rättningen kontrolleras sedan av lärarna och resultatet kungörs samma vecka. Klagomål på rättning av tentan görs till kursledaren. Kursledaren avgör hur och när kompletteringsuppgifter ska redovisas.

Tentaanmälan ska göras.

Varför kamraträttning?

- Det är bra för lärandet att få återkoppling i direkt anslutning till examinationen.
- Att sätta sig in i någon annans lösningar och tankesätt är lärorikt.
- Du får insikt i hur bedömning av tentor går till och vilka överväganden rättande lärare behöver göra.
- Betygsättningen snabbas upp. Resultatet är klart samma dag!

Att tänka på vid kamraträttningen

- Rättnings-sessionen är en obligatorisk del av tentan och vaktas av tentavakter.
- Du bedömer en annan students arbete. Var och en förtjänar en korrekt bedömning. Följ därför rättningsmallen och rättningsanvisningarna så gott du kan.
- Om du är osäker på bedömningen av en uppgift går det bra att fråga. Om du fortfarande är osäker efter frågestunden skriver du ett frågetecken efter eller istället för din bedömning av uppgiften.
- Var saklig och professionell. Gör inte narr av en lösning och skratta inte åt en lösning eller någon annans fråga under kamraträttnings-sessionen.

Muntlig tenta och slutbetyg

Den som fått godkänt på labbarna, båda mästarproven och teoritentan får godkänt på kursen. Slutbetyget bestäms av betygen på samtliga tre betygsatta moment (MAS1, MAS2, LAB1) eventuellt kompletterat med en muntlig tenta och/eller en högrebetygslabb (se tabellen med betygskriterier).

Den som är godkänd på båda mästarproven och har fått minst betyg C på det ena (eller har klarat E- och A-uppgifterna på mästarprov 2) har möjlighet att gå upp på en muntlig tenta för att få högre betyg på mästarproven. Den muntliga tentan kan efter teoritentan bokas in (i Canvas) på tider i tentaveckan i januari 2020. Vid den muntliga tentan kommer läraren att kontrollera att du uppfyller betygskriterierna för det betyg du aspirerar på. Kursböckerna (men inga kompendier eller anteckningar) är tillåtna hjälpmedel.

Slutförande av kursen

Efter period 2 är kursen slut. Den som har labbar kvar att redovisa kan göra det antingen i labbveckan i juni eller på labbpassen i systerkursen DD2352 Algoritmer och komplexitet. Säg till den du redovisar för att du ska ha labben rapporterad på adk19.

Den som har mästarprov kvar kan göra dom i en senare kursomgång av antingen ADK (som går varje hösttermin) eller systerkursen DD2352 (som går varje vårtermin; du ska inte registrera dig på DD2352; du får leta upp kursrummet i Canvas och läsa informationen om mästarproven där). Säg till den du redovisar för att du ska ha mästarprovet rapporterat på adk19.

Den som har tentan kvar kan gå upp på omtentan i april 2020 eller nästa ordinarietenta i december 2020.

När ny kursomgång börjar är det den kursomgångens laborationsuppgifter som gäller. Enskilda laborationer kan tillgodoräknas senare kursomgångar så länge laborationsuppgiften är oförändrad.

Möjlighet till komplettering

Betyget Fx kan kompletteras till E/godkänt för momenten MAS1, MAS2 och TEN1.

Möjlighet till plussning

Plussning är tillåtet för MAS1 och MAS2. För LAB1 kan plussning endast göras från betyg B och C.

Om kursen ändras eller avvecklas

Om provmomenten ändras kommer övergångsbestämmelser i kursplanen att definiera hur den som har kvar gamla provmoment ska examineras.

När kursen inte längre ges har studenten möjlighet att examineras under ytterligare två läsår.

Ytterligare information

Lärplattform

Kursinformation, uppgifter och resultat samlas i [kursrummet i Canvas](#).

Kursen ges av

EECS/Datavetenskap

[Studentexpeditionen](#) ligger i E-huset på Osquars backe 2, plan 4.

Lärare

Kursledare och föreläsare är Viggo Kann viggo@kth.se och Stefan Nilsson snilsson@kth.se.

Det står var och en fritt att välja övningsgrupp eller byta grupp under kursens gång. Övningsassistenter är:

- Jonas Haglund, lite enklare grupp, sal U21
- Lisa Li, normalsvår grupp, sal U31
- Alice Heavey, normalsvår grupp, sal U41
- Marcus Dicander, lite svårare grupp, sal U51

Det finns också ett antal ytterligare personer som är labbhandledare och tar emot mästarprouvsredovisningar.

Kommunikation med lärare

Kontakta gärna lärarna i samband med undervisningen. Använd diskussionsmöjligheten i Canvas för skriftliga frågor.

Kursvärdering och kursanalys

I början av kursen kommer två kursansvariga studenter att utses, som du kan ta kontakt med om du har synpunkter på kursen. Du kan också vända dig direkt till någon av lärarna med synpunkter.

Efter teoritentan kommer en kursenkät att skickas ut till kursdeltagarna. Efter kursens slut kommer kursledarna, i samråd med kursansvariga studenterna, att göra en kursanalys som publiceras på kurswebben.

[Kursanalys för adk18](#)

Tidigare kursanalyser finns upplagda på [kurswebben](#).

Ändringar inför denna kursomgång

Alla föreslagna förändringar i senaste kursanalysen har genomförts. Majoriteten av förändringarna baseras på studentsynpunkter och studentförslag. Sedan föregående kursomgång, adk18, har följande ändrats:

- Gör tydligt att A-uppgiften för mästarpövning 2 inte kan ge annat betyg än A och B. Den som får godkänt på E-uppgiften och A-uppgiften men inte på C-uppgiften ska, om mästarpövning 1 är godkänt, få chans att munta C-kriterierna till betyg A på muntan.
- En text om bitkostnad och enhetskostnad kommer att länkas till från föreläsningen som handlar om detta.
- Som övning på pseudopolynomisk algoritm läggs C-uppgiften från mästarpövning 2 i adk18 in som övningsuppgift.
- Testfall på labb 5 som inte exakt följer labbspecen har uppdaterats.
- Lägg in tips i labblydelsen för labb 4 och 5 om att man kan skriva en egen verifikator för utdata, så att man upptäcker om en otillåten lösning produceras.
- Testa att använda labbredovisningstidsbokning på labbarna (med Remores).
- Använd systemet Remores istället för Doodle för mästarpövningarna.
- Vi beskriver tydligare i början av kursen att lärarnas uppgift är att hjälpa eleverna att klara kursen och att samarbete uppmuntras i alla delar av kursen utom vid mästarpöven.
- Vi beskriver tydligare i början av kursen att teoripoängen som delas ut för labbteori och övningsmästarpövning är ett sätt att klara halva teoritentan redan under kursens gång, att gränsen för godkänt på tentan är hög och att den som kommer mindre än två poäng under godkäntgränsen får komplettera.

Påbyggnad

DD2440 Avancerade algoritmer, DD2442 Seminariekurs i teoretisk datalogi, DD2445 Komplexitetsteori, DD2448 Kryptografins grunder, DD2458 Problemlösning och programmering under press.