HÖGSKOLAN I BORÅS

# Algoritmer och datastrukturer 2
## Algorithms and Data Structures 2
7,5 högskolepoäng
7,5 credits

---

**Ladokkod:** C1AD2B
**Version:** 1.0
**Fastställd av:** Utskottet för utbildningar inom bibliotek, information och IT 2018-12-11
**Gäller från:** VT 2019

**Nivå:** Grundnivå
**Huvudområde (successiv fördjupning):** Datalogi (G1F), Informatik (G1F)
**Utbildningsområde:** Naturvetenskap
**Ämnesgrupp:** Informatik/Data- och systemvetenskap
**Förkunskapskrav:** Avklarade kurser i Programmeringsteknik 7,5 högskolepoäng och Programutveckling – En datalogisk introduktion 7,5 högskolepoäng, eller motsvarande.
**Betygsskala:** Underkänd, Godkänd eller Väl godkänd

---

### Innehåll
Kursens tre huvuddelar är datastrukturer, algoritmteori, samt design och analys av algoritmer. Kursen behandlar träd- och grafstrukturer, och algoritmer som hanterar dessa.

Inom datastrukturer läggs särskilt fokus på hur intern representation påverkar olika operationers effektivitet:
- Träd- och grafstrukturer i C och Python.
- Abstrakta datatyper: träd, grafer, prioritetsköer och symboltabeller

Inom algoritmteori behandlas följande:
- Komplexitetsteori: komplexitetsanalys, rekurrensrelationer, approximativa och heuristiska lösningsstrategier, grundläggande spelteori

Inom design och analys av algoritmer behandlas följande.
- Algoritmer och tillämpningar inom: träd- och graftraversering, oinformerad och informerad sökning, sökträdsbalansering, optimering
- Algoritmtyper: brute force, divide-and-conquer och varianter härav, giriga algoritmer, heuristiska algoritmer, iterative improvement, genetiska algoritmer

Slutligen innehåller kursen färdighetsmoment i programmering, vilket innebär implementering av algoritmer och datastrukturer i C och Python.

### Mål
Efter avklarad kurs ska studenten kunna, med avseende på,

*Kunskap och förståelse*
1.1 redogöra för innebörden och konsekvenserna av komplexitetsklasser,
1.2 redogöra för generiska strategier för algoritmdesign,
1.3 beskriva de träd- och grafstrukturer och abstrakta datatyper som kursen behandlar,
1.4 beskriva etablerade algoritmer för att hantera abstrakta datatyper: träd, grafer, prioritetsköer och symboltabeller
1.5 beskriva etablerade algoritmer för traversering av träd och grafer, samt algoritmer för oinformerad och informerad sökning i träd- och grafstrukturer
1.6 redogöra för grundläggande koncept inom spelteori
1.7 redogöra för heuristiska och approximativa lösningsstrategier
1.8 redogöra för olika implementationer av abstrakta datatyper och algoritmer som arbetar på dessa, i språken C och Python

*Färdighet och förmåga*
2.1 använda matematiska tekniker för vanligt förekommande uppgifter vid analys och design av algoritmer. Specifikt skall studenten kunna sätta upp och lösa rekurrensrelationer för olika typer av operationer på träd- och grafstrukturer,
2.2 utföra komplexitetsanalys av iterativa och rekursiva algoritmer,
2.3 använda och implementera de träd- och grafstrukturer som kursen behandlar, i C och Python
2.4 tillämpa generiska strategier för algoritmdesign,
2.5 utifrån en generell algoritmbeskrivning implementera en algoritm i C och Python,
2.6 implementera och anpassa etablerade algoritmer för hantering av de abstrakta datatyper som kursen behandlar, i C och Python
2.7 designa, genomföra och avrapportera empirisk utvärdering av algoritmer.
2.8 analysera och redogöra för konsekvenser av val av representation, för de abstrakta datatyper och algoritmer som kursen behandlar

*Värderingsförmåga och förhållningssätt*
3.1 välja och utnyttja lämpliga träd- och grafstrukturer vid programkonstruktion,
3.2 välja en lämplig konkret representation för en given abstrakt datatyp

## Undervisningsformer

Undervisning på kursen består av:

- föreläsningar
- övningar
- laborationsseminarier
- handledning

Föreläsningar presenterar kursens teoretiska innehåll. Vid övning arbetar studenter självständigt med uppgifter, under lärares ledning. Handledning stöttar studenter i arbetet med laborationsuppgifter. Vid seminarier presenterar, demonstrerar och diskuterar studenter laborationsuppgifter.

Undervisningen bedrivs på svenska, men undervisning på engelska kan förekomma.

## Examinationsformer

Kursen examineras genom följande examinationsmoment:

- Tentamen: salstentamen
Lärandemål: 1.1 – 1.8, 2.1 – 2.8 och 3.1 – 3.2
Högskolepoäng: 4.5
Betygsskala: Underkänd, Godkänd eller Väl godkänd

- Laboration: Gruppuppgift om träd- och grafstrukturer
Lärandemål: 2.1 – 2.8
Högskolepoäng: 1.0
Betygsskala: Underkänd eller Godkänd

- Inlämningsuppgift: Enskild inlämningsuppgift om algoritmimplementering, -analys och -testning
Lärandemål: 2.1 - 2.8, 3.1 – 3.2
Högskolepoäng: 2.0
Betygsskala: Underkänd eller Godkänd

För betyget Godkänd på hel kurs krävs godkänt betyg på samtliga moment. För betyget Väl Godkänd på hel kurs krävs dessutom Väl Godkänd på *Tentamen: salstentamen*.

Studentens rättigheter och skyldigheter vid examination är enligt riktlinjer och regelverk vid Högskolan i Borås.

## Kurslitteratur och övriga läromedel

Levitin, A. (senaste upplagan), Introduction to the Design and Analysis of Algorithms: International Edition, Pearson Higher Education

Miller, B. N. och Ranum, D.L. (2011), Problem Solving with Algorithms and Data Structures using Python, Franklin

Vetenskapliga artiklar och föreläsningsmaterial kan tillkomma enligt lärarens anvisningar, omfattande max 100 sidor.

### Studentinflytande och utvärdering

Kursen utvärderas i enlighet med gällande riktlinjer för kursvärderingar vid Högskolan i Borås, där studenternas synpunkter ska inhämtas. Kursutvärderingsrapporten publiceras och återkopplas till deltagande och blivande studenter i enlighet med ovan nämnda riktlinjer, och ligger till grund för framtida utveckling av kurser och utbildningsprogram. Kursansvarig lärare ansvarar för att utvärdering enligt ovan genomförs.

### Övrigt

Kursen ingår i Systemarkitekturutbildningen.