

Datastrukturer och algoritmer

Föreläsning 1

1

Innehåll

- Kurspresentation
 - Målsättning
 - Förkunskaper
 - Kursutvärdering
 - Upplägg
 - Översikt
- Föreläsning 1
 - Viktiga begrepp

2

Mina målsättningar

- Alla ska höja sig ett par snäpp som programmerare, och inse att man har gjort det
- Förstå komplexitetsbegreppet
 - Tid och rum
- Ni ska ha lärt er några viktiga typer av algoritmer och datatyper
- Rolig kurs

3

Vad krävs för att nå dit?

- Ge kursen en ärlig chans
 - Kom i tid till föreläsningarna
 - Läs boken, förstå vad som står där (om något är oklart så fråga)
 - Gör övningar
 - Börja med laborationerna i tid
 - Utnyttja handledningen
 - Säg till om det är något som inte funkar

4

Förkunskaper

- Kunna implementera en algoritmbeskrivning i C eller Python

5

Websajten

- <http://www.cs.umu.se/kurser/5DV108/VT10/>
 - Schema
 - Uppdateringar
 - Formalia
 - Resultatredovisning
 - Gruppövningar
 - Föreläsningar
 - OH-bilder
 - Gamla tentor

6

Laborationer

- Tre stycken
 - Fält och lista
 - Analysera tre algoritmer
 - Experimentellt (individuell)
 - Asymptotiskt (gruppövning+ eget arbete)
 - Kortaste vägen
 - Inget särskilt lab är bokad för er så det är fritt fram att använda de obokade labben.

7

Gruppövningar

- Gruppövning 1:
 - Länkad lista
- Gruppövning 2:
 - Komplexitetsanalys (del av lab2)
- Gruppövning 3:
 - Tabeller, stackar och köer
- Gruppövning 4:
 - Träd, grafer och grafalgoritmer
- Gruppövning 5:
 - Sökträd, Trie, Heap, Mängd, Lexikon och Prioritetskö

8

Kursutvärderingen från förra året (delvis annan kurs)

- Önskemål om byte av kursbok.
 - Svårt hitta en bok som håller samma teoretiska klass, och som innehåller det material som täcker kursinnehållet kursen behöver som nuvarande bok.
- Titta igenom ppt-bilderna och se om man kan minska ner dem lite och istället lägga till fler praktiska exempel.
- Plocka tillbaka materialet om AVL-träd (eller andra typer av balancerade träd) som i år uteslöts/gicks igenom väldigt kortfattat.
- Titta över labspecifikationerna och se om de går att göra klarare.

9

Datastrukturer & Algoritmer

- Vi ska strukturera upp ert tänk kring program lite grann
- Genom att se strukturer och hitta likheter ska vi bygga mer komplexa program
- Ni ska lära er lite nya byggblock att bygga upp program av, och lära er i vilken situation de passar

10

Innehållsöversikt

- Algoritmer
 - Komplexitetsanalys, mm
 - Designprinciper
- Abstrakta datatyper
 - Stack, Kö, Listor, Träd, Graf, P-kö. Heap. Tabell, Sträng, Mängd, Sökträd, Tries, mm
- Sökning
 - Olika metoder och hur dom passar i olika lägen
- Sortering
 - 5-6 olika algoritmer, analys

11

Algoritmer

- Tids och rumsfrågor
 - Exekvering, implementering, underhåll
 - Tid vs rumskomplexitet
 - Analys av exekveringstider
 - Olika sätt, vad de står för och +/-
 - Experimentell
 - Asymptotisk
 - $T(n)$ för en alg $\Rightarrow O(n)$
 - Primitiva operationer
 - Okulärbesiktning
 - Att beskriva algoritmer

12

Abstrakta DataTyper

- Beskrivningssätt
 - Organisation, Modell
 - Gränssytan (formell, informell)
- Grundbegrepp
 - Vad är en organisation, sorterad ADT, mm
 - Primär vs. sekundär struktur
 - Absolut vs. relativ komplexitet
- Sätt att implementera
 - +/-, när var hur
 - Komplexitet för de viktigaste operationerna
- Vart hittar vi ADT typiskt
 - Hur olika ADT hänger ihop med varandra
 - Algoritmer på ADT

13

Sökning (traversering)

- Vad är traversering
- Olika metoder för sökning i sekvenser
 - Linjär & Binär
 - +/-
 - Ställer de några krav på implementationen av ADTn

14

Sortering

- Varför sorterar vi?
 - Sorterad ADT vs sortering av data
 - Sökningen blir snabbare...ibland
- Hur kan vi sortera (olika former)
 - +/-
 - När passar dom?
 - Känna till någon bra alg. Bra
 - Stabil sortering

15

Design av algoritmer

- Idéer till 4 olika algoritmtyper
 - Brute force, Divide & Conquer, Greedy och Dynamisk programmering
 - Exempel på varje
 - +/-
 - Typiska användningar

16

Föreläsning 1

- Att bygga program \approx bygga broar
- Brobygge
 - Specifikation
 - Beskrivning
 - Material
 - Verktyg
 - Uppföljning

17

Att bygga program

- Problembeskrivning...
- Systemdesign...
- Modelleringsverktyg
 - **Dat typer** – representera datat i programmet
 - **Algoritmer** – stegvis plan för att utföra något, modellera hur man löser problemet
 - **Kontrollstrukturer** – för att modellera flödet

18

Dat typer = op. + obj.

- Vilka objekt är det?
 - Vad vill vi modellera/representera/abstrahera
- Vad kan man göra med dem?
 - Vilka operationer/metoder

19

Gränssnitt

- Vad är ett gränssnitt?
 - Kontaktyta
 - Gränsen mellan två eller flera delar
 - Överenskommelse - oberoende av vem, vart, hur ska det "passa"
 - Musköten 1700-talet

20

Gränssnitt

- Separerar
 - Funktion och implementationen
 - Användning och skapandet
 - Specifikationen och konstruktionen

21

Gränssnitt i ”datavetenskapen”

- Mellan centralenheter och periferienheter
- Mellan människan och maskinen
- Mellan mjukvarukomponenter
 - Funktioner/metoder
 - Datatyper
 -
 -

22

Begrepp

- Data - objekten som bär information
- Datatyp - ”värdemängd” samt operationer
- Sammansatt datatyp vs. enkel
- Homogen vs. Hetrogen
- Struktur
- Abstrakta datatyper
- Konstruerad datatyp
- Implementerad datatyp
- Fysisk – implementerad i språket/hårdvaran
- Konkret datatyp

23

Beskrivningsätt

- Modell - vardaglig, ”det man modellerar”
 - Kö
- Organisation
 - Den grundläggande naturen på objekten, linjärt ordnade, före och efter relation etc.
- Informell beskrivning
 - Gränsytan (nödvändig och minimal, kraftfull)
 - Informell beskrivning av operationernas funktion
 - Signaturdiagram
- Formell beskrivning - rent matematisk-logisk



24