Essential **C**omputing **II**
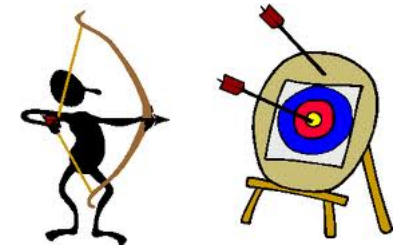
by
Keld Helsgaun

Agenda

- **Purpose of the course**
- **Algorithms and data structures**
- **Prerequisites**
- **Schedule of lectures**
- **Exam**

Purpose

The purpose of this course is to provide a practical introduction to **algorithms** and **data structures** from the viewpoint of abstract thinking and problem solving.

The primary focus is on **problem-solving techniques** that allow the construction of sophisticated time-efficient programs.
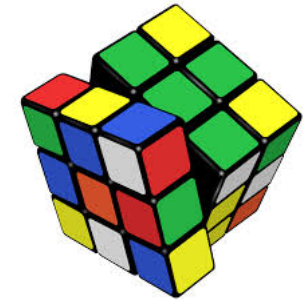
What is an algorithm?

An **algorithm** is a method for solving a problem

Notice that it is not required that an algorithm is executed on a computer

This course deals primarily with computer algorithms

Three important areas of interest:

- **Design**
- **Analysis**
- **Verification**

Design: How do we construct an algorithm?

Analysis: What are the resource demands?

Verification: Can we guarantee correctness?

Why study algorithms?

(1) To make a qualified choice among existing algorithms

(2) To adapt an existing algorithm for a given purpose

(3) To develop new algorithms

Donald Knuth:
*My favorite way to describe computer science is to say that
it is the study of algorithms.*

The data concept

Data:

A *formalized representation* of facts or concepts suitable for communication, interpretation, or processing by people or automated means. **Data on its own carries no meaning**.

Input data → Algorithm → Output data

Information:

The meaning that a human assigns to data by means of known conventions.

What is a data structure?
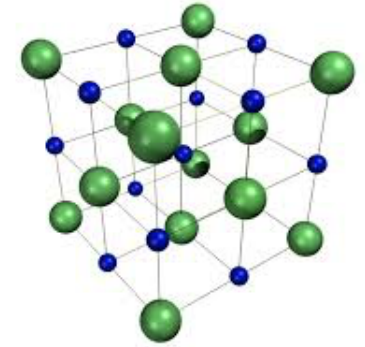
A **data structure** is a particular way of storing and organizing data in a computer so that it can be used efficiently

Examples:
    stack, queue, linked list, tree, hash table, priority queue

Implementation in Java:
    by simple variables, references, arrays, and classes

Prerequisites

Students should have knowledge of either an object-oriented or procedural language. Knowledge of basic features, including primitive data types, operators, control structures, functions (methods), and input and output is assumed.

Knowledge of Java is not assumed.

Schedule of lectures

(1) Preliminaries I

(2) Preliminaries II

        Java in two weeks

(3) Algorithms I

(4) Algorithms II

        Algorithmic design patterns

        Algorithm analysis

(5) Algorithms III

(6) Implementations I

(7) Implementations II

        Data structures

(8) Implementations III

(9) Applications I

        Games, parsing, file compression

(10) Applications II

        Simulation, graphs and shortest paths

Text book

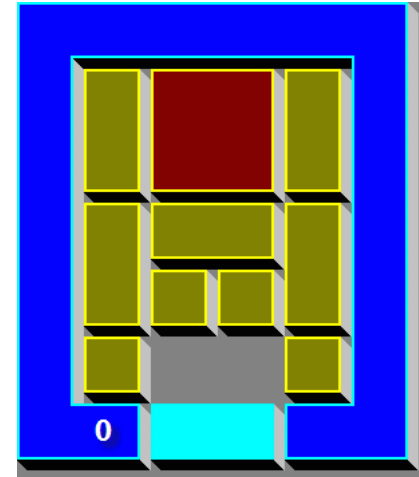**Data Structures & Problem Solving Using Java**

Mark Allen Weiss,

Pearson, 4th edition, 2010.
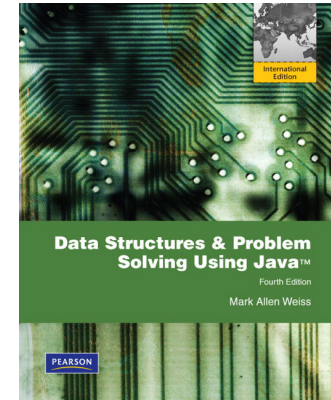
Advantages:

- good explanations in easy-to-read English

- emphasis on data abstraction (hiding away implementation details)

- good examples

- algorithms are expressed in executable code

Exam

Sliding blocks puzzle

15 minutes oral examination based on a practical assignment given during the course.

The exam will focus on the solution of the assignment, and a randomly drawn exam question.

Web page

(available via www.ruc.dk/~keld)

Essential Computing II

Spring 2015

by

Keld Helsgaun

Contents

The course provides knowledge about fundamental algorithms and data structures in computer science. The main objective of this course is to provide students with means to design and reason about algorithms, understand the strengths and weaknesses of known algorithms, and their suitability in particular contexts. The topics covered include data abstraction, generic components, algorithm analysis, recursion, algorithm design, searching, sorting, randomization, simulation, and graphs. In more detail, the contents of the course may be described as follows:

Plan 0
February 2 – February 12

• Read Chapter 1 and 2 in the textbook

Time and place

Lectures and **Exercises**: Thursdays 8^{30}-12^{45}

Room 43-2.29. First lecture: Thursday, February 12