Introduction

Amin Farjudian[1]

Hamlstad University

DT8014 Algorithms Course, Autumn 2016

[1]Originally by Masoumeh Taromirad.

Outline

General Objectives

Deepen and broaden knowledge of

- algorithm analysis
- algorithm design
- advanced classical data structures

With the help of

- overarching concepts
- concrete examples
- mathematical tools
- formal algorithm description

Intended Learning Outcomes

- Knowledge and Understanding
  - explain selected advanced data structures
  - explain concepts of complexity analysis
  - describe selected approaches to dealing with intractable problems
- Skills and Ability
  - given a formal description of an algorithm prove e.g. its correctness
  - formulate an algorithm to solve a given problem using a given technique
  - select appropriate data structures and algorithms for given problems
- Judgement and Approach
  - trade-off between competing aspects when selecting, designing, and implementing data structures and algorithms
  - compare variants of data structures and algorithms on concrete problems

Sessions

1. Lecture Sessions: 2-hour sessions per week
   - lecture and discussion
   - in-class activity or quiz
   - short presentation by students (first half)
     - Each group chooses a topic
   - supervised project (second half)
   - 15-minute BREAK!
2. Lab Sessions: 2 hours per week

Exercises and Project

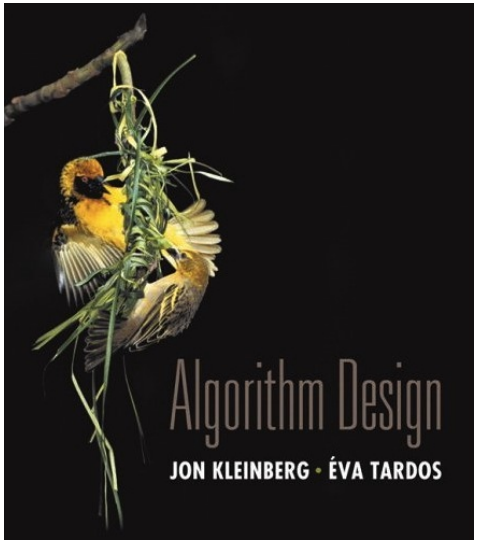
1. Exercises
   - Weekly
   - Deadline: the next lecture
2. Supervised project
   - Second half of the course
   - Pick a project (problem + solution)
   - Weekly brief (oral) report
   - *You* choose the implementation language, the focus is on formal properties of the algorithms.

Examination

- Written project report
- Oral exam

Course Materials

1. Main Textbook
   - Kleinberg, Jon., Tardos, Eva., *Algorithm Design*, Addison Wesly, 2005
2. Other references:
   - Dasgupta, S., Papadimitriou, C., and Vazirani, U., *Algorithms*, McGraw-Hill Education, 2006
3. Slides/Handouts, Exercise, Project
   - on the Blackboard page of the course (hopefully).
   - If not, then on my page
   - updated weekly



Course Material

Recommended Reading Material

- Lecture slides by Kevin Wayne
  - http://www.cs.princeton.edu/~wayne/kleinberg-tardos
- Lectures by Tim Roughgarden (Stanford University)
  - On-line lectures on Coursera
    Algorithms: Design and Analysis, Part 1
    Algorithms: Design and Analysis, Part 2
  - Handouts of Design and Analysis of Algorithms Course (Stanford University)
- Wikipedia

Tentative Schedule

W	**Topic**	**Exercise/Project**
36	Introduction	Ex: Stable Matching
37	Complexity and Graphs	Ex: Hamiltonian Graphs
38	Selected Graph Problems	Ex: Graph Implementation + BFS
39	Greedy Algorithms	Ex: Huffman Trees
40	Randomization	Project Selection (problem+solution)
41	Selected Proof Techniques	Ex: Two Algorithms Pr: Formulate Algorithm
42	Dynamic Programming	Pr: Implement + Analyze
43	Dynamic Programming	Ex: Two DP Problems Pr: Finalization

Visiting exchange professor

- Professor Álvaro Freitas Pereira
  - http://www.inf.ufrgs.br/ afmoreira
- Federal University of Rio Grande do Sul (UFRGS), Brazil
- 3 Weeks: September 4th to September 25th
- Part of the Linnaeus-Palme collaboration between ITE and the Institute of Informatics at UFRGS.
- *A short course on NP-completeness*
- Invaluable opportunity

Any Question?

Break

- Break...

Outline

Travelling Salesman Problem (TSP)

The problem

- a collection of places that need to be visited
- the distance to travel between each pair of places is known.
- **what is the shortest route that visits all of the places exactly once?**

A very practical problem in many contexts

- e.g. courier vehicles choosing routes to deliver parcels, rock bands planning tours, and even a designated driver dropping friends off
- the measurement between each pair does not have to be distance (e.g. the cost to travel between each pair)

Algorithms

- Maths:
  - What is *logically possible*.
- Computer Science:
  - What is *computable*.
    - i. e., for which there exists an *Algorithm*.
- Computational Complexity:
  - What is feasible.

Tractability

- *Tractability* explores problems and algorithms that can take an infeasible amount of time to solve for large enough input sizes.
- A *tractable* problem is one which we can write programs for that finish in a reasonable amount of time.
- An *intractable* problem is one that will generally end up taking way too long (e.g. TSP).
- A very common approach is to give up on getting a perfect answer, and aim for an *approximately* correct answer (in a *heuristic* way).
- P vs NP problem
  - On Clay Mathematics Institute;
  - On Wikipedia.
  - Relation with Moore's law.

Computational Complexity

- An abstraction of the length of time it takes to perform a computation.
- The *actual time* on a particular computer can be useful, but:
  - which computer
  - which language
  - how well-written the program is
  - ...
- Hence, we consider *complexity classes.*
- Example
  - Mergesort: $O(n \log n)$.

In the next sessions ...

We will talk about

- concepts of complexity analysis
  - Asymptotic Complexity, Big-O Notation, Proof Techniques
- advanced data structures
  - Graphs
- approaches to deal with *intractable* problems
  - Greedy Algorithms, Randomization, Dynamic Programming

Outline

Stable Matching Problem

Exercise 1[2]

- *Stable Matching Problem* is a problem of finding a stable matching between two equally sized sets of elements given an ordering of preferences for each element.
  A matching is a mapping from the elements of one set to the elements of the other set.
- Outcome is NOT an efficient algorithm
  - how the problem and a solution can be *formalized*
  - how *hard* it is to check if a suggested solution is stable
  - how many *possible matchings* there are
- Deliverable: written answers should be handed in the next session
- Complete description is available on the course web page

[2] Taken from DT8014, 2014

In this session ...

- How the course is offered (format)
- Why we study algorithms
- What tractability refers to
- What the term *complexity* is used for and why
- What the course is about
- An exercise

Any Question?