COMP 3170 - Analysis of Algorithms & Data Structures

Shahin Kamali

Lecture 1 - Jan. 3, 2018
University of Manitoba

Picture is from the cover of the textbook CLRS.
Introduction
Introduction

In a Glance . . .

- Algorithms are
  - Practical
  - Diverse
  - Fun (really!)
Introduction

In a Glance . . .

- Algorithms are
  - Practical
  - Diverse
  - Fun (really!)

- Let’s ‘learn & play’ algorithms and enjoy
Introduction

Instructor

- B.Sc. in Computer Science (2002-2006)
  - University of Tehran (Iran)
Introduction

Instructor

- B.Sc. in Computer Science (2002-2006)
  - University of Tehran (Iran)
- M.Sc. in Computer Science (2006-2008)
  - Concordia University (Canada)
Introduction

Instructor

- B.Sc. in Computer Science (2002-2006)
  - University of Tehran (Iran)
- M.Sc. in Computer Science (2006-2008)
  - Concordia University (Canada)
- PhD in Computer Science (2008-2014)
  - University of Waterloo (Canada)
Introduction

Instructor

- B.Sc. in Computer Science (2002-2006)
  - University of Tehran (Iran)
- M.Sc. in Computer Science (2006-2008)
  - Concordia University (Canada)
- PhD in Computer Science (2008-2014)
  - University of Waterloo (Canada)
  - LIAFA, Paris (France)
Introduction

Instructor

- B.Sc. in Computer Science (2002-2006)
  - University of Tehran (Iran)
- M.Sc. in Computer Science (2006-2008)
  - Concordia University (Canada)
- PhD in Computer Science (2008-2014)
  - University of Waterloo (Canada)
  - LIAFA, Paris (France)
- Postdoctoral Fellow, Associate (2015-2017)
  - MIT (USA)
Formalities
Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2 Rm: 165 (Jan. 03, 2018 - Apr. 06, 2018)
Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2 Rm: 165 (Jan. 03, 2018 - Apr. 06, 2018)

- Webpage:
Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2 Rm: 165 (Jan. 03, 2018 - Apr. 06, 2018)

- Webpage:
Formalities

Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2 Rm: 165 (Jan. 03, 2018 - Apr. 06, 2018)

- Webpage:

- Piazza:
  https://piazza.com/umanitoba.ca/winter2018/comp3170/home
Formalities

Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2 Rm: 165 (Jan. 03, 2018 - Apr. 06, 2018)

- Webpage:

- Piazza:
  https://piazza.com/umanitoba.ca/winter2018/comp3170/home

- Office hours: 13:00-14:00 on Mondays, 10:30-11:30 am on Tuesdays
  or by appointment

  - You can post your questions (if you prefer anonymously) on Piazza
    so that all your classmates see the solution
  - You should have received an email with respect to Piazza
Formalities

Textbook

- **Required textbook:**

- **Optional textbooks:**
Formalities

Grading

There will be:

- Six assignments (each 5 percent)
- Two quizzes (each 5 percent)
- A midterm exam (20 percent)
- A final exam (40 percent)
Formalities

Grading

There will be:

- Six assignments (each 5 percent)
- Two quizzes (each 5 percent)
- A midterm exam (20 percent)
- A final exam (40 percent)

Theorem

The focus of this course is on learning, practising, and discovering.
Formalities

Grading

There will be:

- Six assignments (each 5 percent)
- Two quizzes (each 5 percent)
- A midterm exam (20 percent)
- A final exam (40 percent)

Theorem

The focus of this course is on learning, practising, and discovering.

Corollary

Having fun in the process is important.
Formalities

Grading (cntd.)

- **Six assignments:**
  - 30 percent of the final mark
  - 10 percent extra for bonus questions.
  - Submit only pdf files (preferably use \LaTeX)
Formalities

Grading (cntd.)

- Six assignments:
  - 30 percent of the final mark
  - 10 percent extra for bonus questions.
  - Submit only pdf files (preferably use \LaTeX).

- Quizzes, Midterm & Final exams:
  - Respectively 10, 20 & 40 percent of the final mark.
  - There will be extra for bonus questions in midterm and final.
  - All are closed-book.
  - Sample exams will be provided for practice for midterm and final.
Important Dates (tentative)

Important Dates

January 3 first class
January 22 assignment 1 due
January 31 quiz
February 9 assignment 2 due
February 19 assignment 3 due
February 20–23 midterm break - no class
February 28 midterm exam
March 9 assignment 4 due
March 16 voluntary withdrawal deadline
March 19 assignment 5 due
March 14 Pi (π or pie) day
March 28 quiz
March 30 Good Friday - no class
April 4 assignment 6 due
April 6 last class
April 9–23 exam period
Prerequisites

- What I have learned from previous courses?
- Basic sorting algorithms, e.g., quick sort and merge sort
Prerequisites

- What I have learned from previous courses?
- Basic sorting algorithms, e.g., quick sort and merge sort
- Asymptotic notations, e.g., big $O$, $\Omega$, etc.
Basic Concepts

Prerequisites

- What I have learned from previous courses?
- Basic sorting algorithms, e.g., quick sort and merge sort
- Asymptotic notations, e.g., big $O$, $\Omega$, etc.
- Basic abstract data types (ADTs) and data structures
  - Stacks, queues, dictionaries, binary search trees, hash tables, graphs.
Basic Concepts

Prerequisites

- What I have learned from previous courses?
- Basic sorting algorithms, e.g., quick sort and merge sort
- Asymptotic notations, e.g., big $O$, $\Omega$, etc.
- Basic abstract data types (ADTs) and data structures
  - Stacks, queues, dictionaries, binary search trees, hash tables, graphs.
- The difference between an ADT and a data structure? (wait for two slides)
Basic Concepts

Prerequisites

- What I have learned from previous courses?
- Basic sorting algorithms, e.g., quick sort and merge sort
- Asymptotic notations, e.g., big $O$, $\Omega$, etc.
- Basic abstract data types (ADTs) and data structures
  - Stacks, queues, dictionaries, binary search trees, hash tables, graphs.
- The difference between an ADT and a data structure? (wait for two slides)
- Basic algorithm families
  - Greedy algorithms, dynamic programming algorithms, divide & conquer (d&c)
Basic Concepts

Prerequisites

- What I have learned from previous courses?
- Basic sorting algorithms, e.g., quick sort and merge sort
- Asymptotic notations, e.g., big $O$, $\Omega$, etc.
- Basic abstract data types (ADTs) and data structures
  - Stacks, queues, dictionaries, binary search trees, hash tables, graphs.
- The difference between an ADT and a data structure? (wait for two slides)
- Basic algorithm families
  - Greedy algorithms, dynamic programming algorithms, divide & conquer (d&c)
- Analysis techniques
  - E.g., how to analyse time complexity of a d&c algorithm?
Basic Concepts

Prerequisites

- What I have learned from previous courses?
- Basic sorting algorithms, e.g., quick sort and merge sort
- Asymptotic notations, e.g., big $O$, $\Omega$, etc.
- Basic abstract data types (ADTs) and data structures
  - Stacks, queues, dictionaries, binary search trees, hash tables, graphs.
- The difference between an ADT and a data structure? (wait for two slides)
- Basic algorithm families
  - Greedy algorithms, dynamic programming algorithms, divide & conquer (d&c)
- Analysis techniques
  - E.g., how to analyse time complexity of a d&c algorithm?
  - Solving recursions, Master theorem, etc.
Algorithms
Basic Concepts

Algorithms

- What is an algorithm?
Algorithms

What is an algorithm?

**Definition**

An algorithm is a computational procedure formed by a sequence of instructions (steps) to solve a problem.
Algorithms

What is an algorithm?

Definition
An algorithm is a computational procedure formed by a sequence of instructions (steps) to solve a problem.

- The problem has an input and often requires an output.
What is an algorithm?

**Definition**

An algorithm is a computational procedure formed by a sequence of instructions (steps) to solve a problem.

- The problem has an **input** and often requires an **output**.
- Transition from one step to another can be **deterministic** or **randomized**.
  - The algorithm is deterministic if it never uses randomization; otherwise, it is a randomized algorithm.
What is an algorithm?

**Definition**

An algorithm is a computational procedure formed by a sequence of instructions (steps) to solve a problem.

- The problem has an **input** and often requires an **output**
- Transition from one step to another can be **deterministic** or **randomized**
  - The algorithm is deterministic if it never uses randomization; otherwise, it is a randomized algorithm
- Solving the problem requires the algorithm to **terminate**.
  - **Time complexity** concerns the number of steps that it takes for the algorithm to terminate (often on the worst-case input)
Basic Concepts

Abstract Data Type

What is an Abstract Data Type (ADT)

Definition

An abstract data type is formed by I) a set of values (data items) and II) a set of operations allowed on these items

- Stack is an ADT. Data items can be anything and operations are *push* and *pop*
- An ADT is abstract way of looking at data (no implementation is prescribed)
- An ADT is the way data ‘looks’ from the view point of user
Basic Concepts

Data Structure

- What is a Data Structure

**Definition**

A data structure is a concrete representation of data, including how data is organized, stored, and accessed on a computer.

- A linked-list is a data structure
- Data structures are *implementations* of ADTs
- A data structure is the way data ‘looks’ from the view point of implementer
ADTs vs Data Structures

- ADTs: Stacks, queues, priority queues, dictionaries
- Data structures array, linked-list, binary-search-tree, binary-heap, hash-table-using-probing, hash-table-using-chaining, adjacency list, adjacency matrix, etc.
Topics

Course topics

asymptotic notations (review)
selection
balanced search trees
skip lists
binomial heaps
amortized analysis
disjoint sets

lower bounds
computational complexity
approximation algorithms
graph algorithms
randomized algorithms
string matching