COMP 3170 - Analysis of Algorithms & Data Structures

Shahin Kamali

Topic 1 - Introductions
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)
- Assistant professor (since 2017)
  - University of Manitoba
Formalities
Formalities

Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2  **Rm: 160** (Jan. 07, 2018 - Apr. 09, 2018)
Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2 Rm: 160 (Jan. 07, 2018 - Apr. 09, 2018)

- Webpage:
  http://www.cs.umanitoba.ca/~kamalis/winter19/comp3170.html
Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2 **Rm: 160** (Jan. 07, 2018 - Apr. 09, 2018)

- Webpage:
Logistics

- **Lecture:** Mondays, Wednesdays, and Fridays, 10:30-11:20am EITC E2 Rm: 160 (Jan. 07, 2018 - Apr. 09, 2018)

- **Webpage:**
  http://www.cs.umanitoba.ca/~kamalis/winter19/comp3170.html

- **Piazza:** piazza.com/umanitoba.ca/winter2019/comp3170
Logistics

- Lecture: Mondays, Wednesdays, and Fridays, 10:30-11:20am
  EITC E2 Rm: 160 (Jan. 07, 2018 - Apr. 09, 2018)
- Webpage:
  http://www.cs.umanitoba.ca/~kamalis/winter19/comp3170.html
- Piazza: piazza.com/umanitoba.ca/winter2019/comp3170
- Office hours: 1:00 to 2:00 pm on Mondays, 10:30 to 11:30 am
  Tuesdays, Location: E2 586 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

- The main reference:

- Optional textbooks:
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.
Grading (cntd.)

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

January 7 first class
January 23 assignment 1 due
January 30 quiz
February 8 assignment 2 due
February 18 assignment 3 due
February 19–22 midterm break - no class
March 2 midterm exam
March 8 assignment 4 due

March 14 Pi (π or pie) day
March 20 voluntary withdrawal deadline
March 22 assignment 5 due
March 29 quiz
April 5 assignment 6 due
April 9 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?
Basic Concepts

Algorithms

What is an algorithm?

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

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**
Basic Concepts

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.
- 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).
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
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.
Basic Concepts

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.
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