Lecture 01/09/17

Lecturer: Xiaodi Wu

Reading Assignment: Course Website, Chapter 1, 2.1, 2.2
Welcome to CIS 313: Intermediate Data Structure
Welcome to CIS 313: Intermediate Data Structure
&
Happy new year!
Teaching Team

Instructor

▶ Instructor: Prof. Xiaodi Wu
▶ Contact: DES 332, xiaodiwu@cs.uoregon.edu
▶ Research: Quantum Computation
▶ CIS 410/510 Introduction to Quantum Information Theory, Spring 2017.
Teaching Team

Instructor

- Instructor: Prof. Xiaodi Wu
- Contact: DES 332, xiaodiwu@cs.uoregon.edu
- Research: Quantum Computation

GTFs

- Samuel Mergendahl, smergend@cs.uoregon.edu, DES 232
  Labs: M 13:00 - 13:50, F 10:00 - 10:50
- Zhangxiang Hu, huz@cs.uoregon.edu, DES 232
  Labs: R 12:00 - 12:50
More logistics

Office Hours on every weekday

▶ Wu: W 12:30pm - 2:00pm, F 12:30pm - 2:00pm at DES 332
▶ Mergendahl: M 2:00pm - 4:00pm, R 9:00am - 11:00am at DES 232
▶ Hu: T 10:00am - 12:00am at DES 232
More logistics

Office Hours on every weekday

▶ Wu: W 12:30pm - 2:00pm, F 12:30pm - 2:00pm at DES 332
▶ Mergendahl: M 2:00pm - 4:00pm, R 9:00am - 11:00am at DES 232
▶ Hu: T 10:00am - 12:00am at DES 232

Websites

▶ Course website: syllabus, reading assignments, lecture notes (posted before and modified after lectures), and so on. Check Frequently!!.
More logistics

Office Hours on every weekday

▶ Wu: W 12:30pm - 2:00pm, F 12:30pm - 2:00pm at DES 332
▶ Mergendahl: M 2:00pm - 4:00pm, R 9:00am - 11:00am at DES 232
▶ Hu: T 10:00am - 12:00am at DES 232

Websites

▶ **Course website**: syllabus, reading assignments, lecture notes (posted before and modified after lectures), and so on. Check **Frequently!!**.
▶ **Piazza**: announcements, discussion forum, ask for helps.
More logistics

Office Hours on every weekday

- Wu: W 12:30pm - 2:00pm, F 12:30pm - 2:00pm at DES 332
- Mergendahl: M 2:00pm - 4:00pm, R 9:00am - 11:00am at DES 232
- Hu: T 10:00am - 12:00am at DES 232

Websites

- **Course website**: syllabus, reading assignments, lecture notes (posted before and modified after lectures), and so on. Check Frequently!!
- **Piazza**: announcements, discussion forum, ask for helps.
- **Canvas**: distribute and submit Lab assignments, grades, solutions.
Evaluation

▶ **Active participation (3%)** in lectures, labs, and office hours in strongly encouraged and rewarded. **Bonus:** (2%) for actively helping other students (make us know).
Evaluation

- **Active participation (3%)** in lectures, labs, and office hours in strongly encouraged and rewarded. **Bonus: (2%)** for actively helping other students (make us know).

- **Homework and Labs (42%)**: 4 written assignments and 5 lab assignments. 5% each except Lab 0 (2%). **Bonus: (10%)** for bonus problems in both assignments and labs.

- **Exams**: **midterm (25%)** tentative date 02/08/17, **final (30%)** 03/23/17. One review lecture for each exam.
Evaluation

- **Active participation (3%)** in lectures, labs, and office hours in strongly encouraged and rewarded. **Bonus:** (2%) for actively helping other students (make us know).

- **Homework and Labs (42%)**: 4 written assignments and 5 lab assignments. 5% each except Lab 0 (2%). **Bonus:** (10%) for bonus problems in both assignments and labs.

- **Exams**: **midterm (25%)** tentative date 02/08/17, **final (30%)** 03/23/17. One review lecture for each exam.

- **Grading**: B- or above if your grade is above the average/median of the class. **Canvas Score is NOT accurate.**

- **Late Policy**: We have a STRICT late policy. **NO acceptance once sample solutions are posted or in the exam week.** i.e., please submit by Mar 19th, 2017 for everything.
Labs

What are they about?

- Implementing (Java) data structures from the class!
Labs

What are they about?

- Implementing (Java) data structures from the class!

Schedule

- Lab 0 (one-week lab in Week 1) setting up the format, due 01/15/17.
Labs

What are they about?

▶ Implementing (Java) data structures from the class!

Schedule

▶ Lab 0 (one-week lab in Week 1) setting up the format, due 01/15/17.
▶ Lab 1, 2, 3, 4 (two-week labs). 1st week: description of the lab; 2nd week: help with the lab and the assignment.
Labs

What are they about?

▶ Implementing (Java) data structures from the class!

Schedule

▶ Lab 0 (one-week lab in Week 1) setting up the format, due 01/15/17.
▶ Lab 1, 2, 3, 4 (two-week labs). 1st week: description of the lab; 2nd week: help with the lab and the assignment.
▶ No lab in Week 10. Replaced by office hours for the final.
Question

What is your motivation to study this course?
Question

What is your motivation to study this course?

- major requirement; fundamental skills for coders; and for coding interviews (Google; Facebook; Microsoft; ...)

CIS 313 is about:
- Tools and methods to analyze programs (codes): prove its correctness, determine its performance, ......
- in the context of fundamental data structures. Get familiar with data structures as well!

CIS 315 is in the context of fundamental algorithms.

Hard course: math maturity (proofs) and coding skills!!
Question

What is your motivation to study this course?

- major requirement; fundamental skills for coders; and for coding interviews (Google; Facebook; Microsoft; ...)

CIS 313 is about:

- Tools and methods to analyze programs (codes): prove its correctness, determine its performance, ......
Question

What is your motivation to study this course?

▶ major requirement; fundamental skills for coders; and for coding interviews (Google; Facebook; Microsoft; ...)

CIS 313 is about:

▶ Tools and methods to analyze programs (codes): prove its correctness, determine its performance, ......

▶ in the context of fundamental data structures. Get familiar with data structures as well!
Question

What is your motivation to study this course?

► major requirement; fundamental skills for coders; and for coding interviews (Google; Facebook; Microsoft; ...)

CIS 313 is about:

► Tools and methods to analyze programs (codes): prove its correctness, determine its performance, ......
► in the context of fundamental data structures. Get familiar with data structures as well!
► CIS 315 is in the context of fundamental algorithms.
Question

What is your motivation to study this course?

- major requirement; fundamental skills for coders; and for coding interviews (Google; Facebook; Microsoft; ...)

CIS 313 is about:

- Tools and methods to analyze programs (codes): prove its correctness, determine its performance, ......
- in the context of fundamental data structures. Get familiar with data structures as well!
- CIS 315 is in the context of fundamental algorithms.
- **Hard course:** math maturity (proofs) and coding skills !!.
Analysis of programs

Experimental Study: drawbacks
▶ Limited input cases; need good test data.
▶ Hard to compare efficiency between algorithms due to hardware and software environments.
▶ One needs to do the coding!

Ideal Analytical Framework
▶ Take into account all possible cases.
▶ Compare efficiency between algorithms independent of hardware and software environments.
▶ Need only high-level descriptions.
Analysis of programs

Experimental Study: drawbacks

- Limited input cases; need good test data.
- Hard to compare efficiency between algorithms due to hardware and software environments.
- One needs to do the coding!

Ideal Analytical Framework

- Take into account all possible cases.
- Compare efficiency between algorithms independent of hardware and software environments.
- Need only high-level descriptions.
Analysis of programs

Experimental Study: drawbacks

- Limited input cases; need good test datas.
- Hard to compare efficiency between algorithms due to hardware and software environments.

Ideal Analytical Framework

- Take into account all possible cases.
- Compare efficiency between algorithms independent of hardware and software environments.
- Need only high-level descriptions.
Analysis of programs

Experimental Study: drawbacks

- Limited input cases; need good test datasets.
- Hard to compare efficiency between algorithms due to hardware and software environments.
- One needs to do the coding!
Analysis of programs

Experimental Study: drawbacks

- Limited input cases; need good test data.
- Hard to compare efficiency between algorithms due to hardware and software environments.
- One needs to do the coding!

Ideal Analytical Framework

- Take into account all possible cases.
Analysis of programs

Experimental Study: drawbacks

- Limited input cases; need good test data.
- Hard to compare efficiency between algorithms due to hardware and software environments.
- One needs to do the coding!

Ideal Analytical Framework

- Take into account all possible cases.
- Compare efficiency between algorithms independent of hardware and software environments.
Analysis of programs

Experimental Study: drawbacks

▶ Limited input cases; need good test datas.
▶ Hard to compare efficiency between algorithms due to hardware and software environments.
▶ One needs to do the coding!

Ideal Analytical Framework

▶ Take into account all possible cases.
▶ Compare efficiency between algorithms independent of hardware and software environments.
▶ Need only high-level descriptions.
Example: Insertion-Sort (A) on page 18, Chapter 2

**Algorithm:** Insertion-Sort (A):

**Input:** An array A storing \( n \geq 1 \) integers.

**Output:** A sorted array (in non-decreasing order) A.

```plaintext
for j ← 2 to A.length do
    key = A[j].
    i = j - 1.
    while i > 0, A[i] > key do
        i = i - 1.
    end while
    A[i + 1] = key.
end for
```
Comparisons: modeling of computation costs

Primitive Operations: Unit 1

- Assigning a value to a variable; Indexing into an array; Comparing two numbers;
- An arithmetic operation (+); Calling and returning from a method; ......
Comparisons: modeling of computation costs

Primitive Operations: Unit 1

- Assigning a value to a variable; Indexing into an array; Comparing two numbers;
- An arithmetic operation (+); Calling and returning from a method; ......

Pros and Cons

Pros: only need to count the number of primitive operations.
Comparisons: modeling of computation costs

Primitive Operations: Unit 1

- Assigning a value to a variable; Indexing into an array; Comparing two numbers;
- An arithmetic operation (+); Calling and returning from a method; ......

Pros and Cons

Pros: only need to count the number of primitive operations.

Cons: model-specific (input encoding, OS, ...) ; exact number is too labor-intense and unnecessary.
Comparisons: modeling of computation costs

**Primitive Operations: Unit 1**

- Assigning a value to a variable; Indexing into an array; Comparing two numbers;
- An arithmetic operation (+); Calling and returning from a method; ......

**Pros and Cons**

**Pros**: only need to count the number of primitive operations.

**Cons**: model-specific (input encoding, OS, ...) ; exact number is too labor-intense and unnecessary.
Worst case vs Average Case

Worst case analysis: adopted in this course

- Maximum number of primitive operations over all cases.
- Hardest case! Usually easier to analyze than average case study.
Worst case vs Average Case

Worst case analysis: adopted in this course

- Maximum number of primitive operations over all cases.
- Hardest case! Usually easier to analyze than average case study.
- Worst cases might be rare in practice.
Worst case vs Average Case

Worst case analysis: adopted in this course

- Maximum number of primitive operations over all cases.
- Hardest case! Usually easier to analyze than average case study.
- Worst cases might be rare in practice.

Average case analysis

- Average number of primitive operations over a distribution of all cases.
- Average performance. Harder to analyze. Quality depends on the distribution.
Worst case vs Average Case

Worst case analysis: adopted in this course

- Maximum number of primitive operations over all cases.
- Hardest case! Usually easier to analyze than average case study.
- Worst cases might be rare in practice.

Average case analysis

- Average number of primitive operations over a distribution of all cases.
- Average performance. Harder to analyze. Quality depends on the distribution.
Asymptotic Notation

Goals

- Insensitive to constant factors: (1) different models generate constant factors (2) no capture of asymptotic behavior.

Definition (Big-Θ)

For any \( g(n) : N \rightarrow \mathbb{R}^+ \), we denote by \( \Theta(g(n)) \) the set of functions \( \Theta(g(n)) = \{ f(n) : \exists c_1, c_2, n_0, s.t. 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0 \} \).
Asymptotic Notation

Goals

▶ Insensitive to constant factors: (1) different models generate constant factors (2) no capture of asymptotic behavior.
▶ Capture dominating terms and simplify analysis.
Asymptotic Notation

Goals

▶ Insensitive to constant factors: (1) different models generate constant factors (2) no capture of asymptotic behavior.
▶ Capture dominating terms and simplify analysis.
▶ Demonstrate asymptotic behavior. A guideline for determining efficiency (See Chapter 3).

Definition (Big-Θ)

For any \( g(n) : \mathbb{N} \rightarrow \mathbb{R}^+ \), we denote by \( \Theta(g(n)) \) the set of functions \( \Theta(g(n)) = \{ f(n) : \exists c_1, c_2, n_0, \text{s.t.} 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0 \} \).
Asymptotic Notation

Goals

- Insensitive to constant factors: (1) different models generate constant factors (2) no capture of asymptotic behavior.
- Capture dominating terms and simplify analysis.
- Demonstrate asymptotic behavior. A guideline for determining efficiency (See Chapter 3).
- **Cons**: huge constants impact practical cases (non-asymptotic cases)

**Definition (Big-Θ)**

For any \( g(n) : N \rightarrow R^+ \), we denote by \( \Theta(g(n)) \) the set of functions \( \Theta(g(n)) = \{ f(n) : \exists c_1, c_2, n_0, s.t., 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0 \} \).
Asymptotic Notation

Goals

- Insensitive to constant factors: (1) different models generate constant factors (2) no capture of asymptotic behavior.
- Capture dominating terms and simplify analysis.
- Demonstrate asymptotic behavior. A guideline for determining efficiency (See Chapter 3).
- **Cons**: huge constants impact practical cases (non-asymptotic cases)

Definition (Big-Θ)

For any \( g(n) : \mathbb{N} \rightarrow \mathbb{R}_+ \), we denote by \( \Theta(g(n)) \) the set of functions

\[
\Theta(g(n)) = \{ f(n) : \exists c_1, c_2, n_0, \text{s.t.}, 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq n_0 \}
\]