CSCI 338 — Algorithm Design and Analysis

Jeremy Iverson

College of Saint Benedict & Saint John's University

instructor

- Jeremy Iverson (`jiverson002@csbsju.edu`)
- MAIN 258, (320) 363-5542
- office hours: T   10:00   -   11:00
                  W   14:00   -   15:00

textbook

- *Introduction to the Design and Analysis of Algorithms*, 3rd Edition, Levitin

website

- `https://csbsju.instructure.com/courses/13537`

- I prefer to be called Jeremy
- Encourage questions right away
- Emphasize the importance of the Canvas site for finding information about the class
- office hours
  - Mention outlook calendar & my home page
    - For those unfamiliar with Outlook meetings, then they should schedule another way and we will go over this in meeting
- go through Canvas page organization quickly

objectives

to use different abstract methodologies to construct algorithms that solve given problems

to analyze the time and space complexity of algorithms and compare the complexity of different algorithms

to describe the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ and explain why they are important in understanding computational tractability

to write a good description and analysis of an algorithm

objectives

to use different abstract methodologies to construct algorithms that solve given problems

to analyze the time and space complexity of algorithms and compare the complexity of different algorithms

to describe the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ and explain why they are important in understanding computational tractability

to write a good description and analysis of an algorithm

2

objectives

to use different abstract methodologies to construct algorithms
that solve given problems

**to analyze the time and space complexity of algorithms and
compare the complexity of different algorithms**

to describe the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ and explain why
they are important in understanding computational tractability

to write a good description and analysis of an algorithm

objectives

to use different abstract methodologies to construct algorithms that solve given problems

to analyze the time and space complexity of algorithms and compare the complexity of different algorithms

to describe the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ and explain why they are important in understanding computational tractability

to write a good description and analysis of an algorithm

objectives

to use different abstract methodologies to construct algorithms that solve given problems

to analyze the time and space complexity of algorithms and compare the complexity of different algorithms

to describe the complexity classes $\mathcal{P}$ and $\mathcal{NP}$ and explain why they are important in understanding computational tractability

to write a good description and analysis of an algorithm

daily written assignments

three (3) programming assignments

- do your own work
- use each other as resources only
- due dates are strict (no partial credit for late assignments)

three (3) topical exams and one (1) cumulative

- closed book / open note

point distribution

- written assignments: 24% total
- programming assignments: 4% each
- exams: 13% each
- final: 25%

Should be student's own work, may work together but must individual solutions

Want to encourage using each other as resources, but do not want some students to rely on other students for answers

need more info...

see `course syllabus`!

develop a framework for analyzing algorithms

study some of the fundamental algorithm design techniques

look at a sample of the important algorithms in CS

a look ahead

develop a framework for analyzing algorithms

study some of the fundamental algorithm design techniques

look at a sample of the important algorithms in CS

we need to be able to reason about the algorithms that we design. we need some way to compare different algorithms that solve the same problem to understand which is better and when...insertion sort is easy to understand and implement, however, it is only the right best sorting algorithm under very specific circumstances.

develop a framework for analyzing algorithms

study some of the fundamental algorithm design techniques

look at a sample of the important algorithms in CS

algorithm design is not a process that starts from a blank slate every time — complex problem-solving often relies on building blocks or reformulating one problem as another — we will be learning some of these building blocks

1. which is better and why, insertion sort or merge sort?

activity

1. which is better and why, insertion sort or merge sort?

2. is there any circumstance when bubble sort is better than merge sort?

a bit of parting advice

remember that this is a 300-level course

I expect you to do some learning on your own and come to class prepared to enhance / clarify / correct the understanding the you created. I want to help you explore your understanidng of the material, not necessarily help you gain a basic understanding of the material

a bit of parting advice

remember that this is a 300-level course

if something is confusing, tell me

I expect you to do some learning on your own and come to class prepared to enhance / clarify / correct the understanding the you created. I want to help you explore your understanidng of the material, not necessarily help you gain a basic understanding of the material

otherwise, I am relying on my own experience to decide which material to emphasize and/or clarify

teaching philosophy

Your job is to empower those you teach; when you do for them what they should be doing for themselves, you create dependency rather than empowerment.

It is easy to give in to the frustration that results from seeing amazing possibilities for the people you are teaching, and you want it more for them than they want it for themselves.

Don't give in to that frustration!
— Based on passage from "Resisting Happiness" by Matthew Kelly

I sincerely believe that all of you can master this material and I want more than anything else this semester, to help you do that!However, don't expect me to give you answers. And do get frusterated with me when I challenge you to discover the answers for yourself.

questions?