CSCI 338 — Algorithm Design and Analysis

Jeremy Iverson

College of Saint Benedict & Saint John's University

logistics

instructor

- Jeremy Iverson (`jiverson002@csbsju.edu`)
- PENGL 213, (320) 363-3083
- office hours: TBD

textbook

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

website

- `https://csbsju.instructure.com/courses/11420`

- 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 and 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 and 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 and 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 and 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 and and explain why they are important in understanding computational tractability

to write a good description and analysis of an algorithm

evaluation

eight (8) 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: 3% each
- 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`!

a good question

why bother?

- better debugger
- better programmer
- better problem-solver

most of you will neither write nor maintain compilers nor will you program in assembly on a regular basis, so then why bother...

- better debugger — improve tracing skill and have a better idea of how things you write are understood by the machine
- better programmer — debugging is a skill that many “programmers” lack, but you will have improved this skill therefore, you will be a better programmer
- better problem-solver — because of your new understanding of the orgranization of a computer system you will better understand its capabilities and limitations — you will also gain experience moving between levels of abstraction, thus improving your abstract thinking

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

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...bubble sort is easy to understand and implement, however, it is only the right best sorting algorithm under very specific circumstances.

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

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

activity

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

activity

1. which is better and why, bubble 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?

activity

download this slide deck and follow the instructions on the next slide

if there is still time, talk about all of the material for this course being hosted on GitHub, and how they can access it