Lab 1: Intro to Lab

Monday, September 3
CompSci 531, Fall 2018
Outline

• Purpose and Logistics of Lab

• Math and Programming: what you should already know

• \LaTeX: How to type mathematics

• Insertion Sort vs. Mergesort in practice
Purpose of Lab

• “Lectures will focus on introducing major algorithmic principles of design and analysis, along with mathematical analysis of algorithmic problems. The weekly lab section will build on that material to explore questions of implementations and applications to real world problems.”

• In lab, we will work more with:
  • Data analysis
  • Implementation
  • (But still math!)
Logistics of Lab

• Website:  www2.cs.duke.edu/courses/fall18/compsci531/index

• Meet weekly at 11:45 am to 1:00 pm in LSRC B101

• Usually I will introduce and explain a topic, then ask you to work on a demo in small groups.

• 4 lab homework assignments
  • assignments will be released and submitted through Sakai.
  • complete in small groups (1-3 students). Turn in one write up with all of your names. You will all receive the same grade.
  • assignments will usually involve some programming and playing with data, and you will turn in code and a written report.
  • you may not share your work outside of your group, and may not use code or materials from the internet.
Outline

• Purpose and Logistics of Lab

• Math and Programming: what you should already know

• LaTeX: How to type mathematics

• Insertion Sort vs. Mergesort in practice
Math: What You Should Already Know

• Algebra: logarithms/exponentials, manipulating finite sequences and summations, systems of equations and inequalities.

• Counting/Probability: events, inclusion/exclusion, pigeonhole principle, conditional probability, random variables, and expectations.

• Logic/Proofs: boolean formulas, quantifiers, set theory, proof by contradiction, weak & strong induction, and writing proofs in English.

• Linear Algebra: basic definitions and matrix multiplication.

• Graph Theory: basic definitions, degrees, isomorphisms, bipartite graphs, and matchings.

• More information: Check out a recent iteration of a CompSci 230. If you need to brush up, Mathematics for Computer Science is a good reference.
Programming: What You Should Already Know

• Basic Programming: iterative and recursive functions, basic data structures (lists, queues, stacks, etc), and file input/output.

• Languages:
  • Preferred: Java, c, c++, Python, Matlab or R.
  • Other: If you aren’t comfortable with any of the above languages, please contact me.
  • For generating figures, plots, etc, you can use excel if you prefer.

• More info: Check out a recent iteration of CompSci 201 (the class is taught in java, but other languages are fine).

• Reiteration: You may not share code outside of your groups. I may use software to detect plagiarism, and take this very seriously.
Outline

• Purpose and Logistics of Lab

• Math and Programming: what you should already know

• LaTeX: How to type mathematics

• Insertion Sort vs. Mergesort in practice
LaTeX: How to type mathematics

• “It is strongly encouraged that students should type their solutions using LaTeX...If handwritten solutions are illegible, they will not be graded.”

• LaTeX is the preeminent markup language for writing well formatted mathematics.

• Works like html – you write a source .tex document that is mostly text and equations, with commands specified with a /. The .tex is compiled into a .pdf for viewing.

• You find free distributions of LaTeX from latex-project.org/get or you can use a free web based editor like sharelatex.com or overleaf.com.
LaTeX: How to type mathematics

• Let’s look at a brief demo LaTeX document.
• This demo document is available under resources on Sakai.
• More resources and references:
  • A short introduction translated into many languages: ctan.org/pkg/lshort.
  • Tutorials for learning LaTeX sharelatex.com/learn.
  • LaTeX cheat sheet (2 page reference): https://wch.github.io/latexsheet/
Outline

• Purpose and Logistics of Lab

• Math and Programming: what you should already know

• LaTeX: How to type mathematics

• Insertion Sort vs. Mergesort in practice
Insertion Sort vs. Mergesort

• In lecture last Thursday, Aug. 30, we talked about asymptotic analysis of insertion sort and mergesort.

• We concluded: “…merge sort asymptotically beats insertion sort in the worst case. In practice, merge sort beats insertion sort for n > 30 or so. Go test it out for yourself!”

• Let’s do this!
Insertion Sort vs. Mergesort

• Break into groups of 3 or 4.

• Code up insertion sort and mergesort. You can find the pseudocode from the lecture slides off the course website: cs.duke.edu/courses/fall18/compsci531

• Download lab1.zip from the website. There are 4 text files, each with a list of numbers: two with 1000 entries and two with 5000 entries.

• Use your system timer to check how long your sorts take.

• Explain your results.
Conclusion

• Knowing your data matters!
• You will get to play around a bit more with sorting algorithms in your first lab assignment.
• Next week, we will look at Strassen’s multiplication algorithm.