ECE360C: Algorithms
Course Logistics

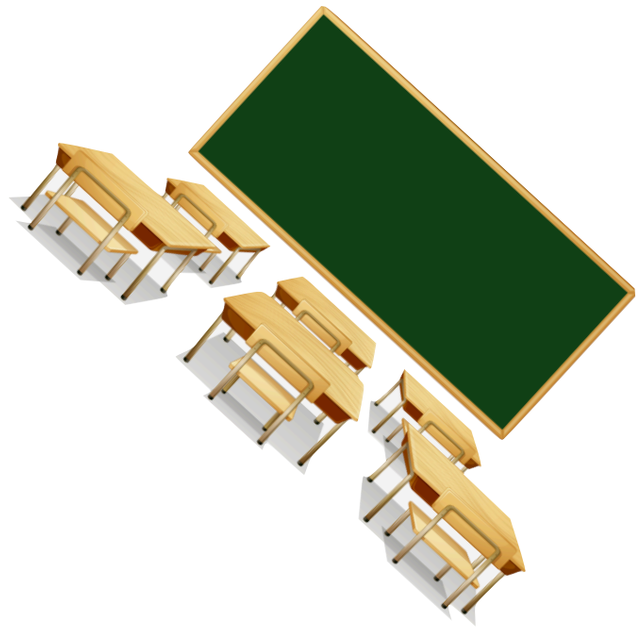
Dr. David Soloveichik

Department of Electrical and Computer Engineering
University of Texas at Austin

Prerequisites

- **Discrete Math** and **ECE312 Software Design and Implementation I**
- You should be very comfortable with basic proof techniques such as proof by induction, contradiction, etc, as well as basic mathematical objects such as sets, graphs, etc.
- You should be comfortable writing, compiling, and debugging programs of a moderate complexity (i.e., hundreds of lines of code). Course programming will be done in Java; lectures will not include instruction in any programming language, but preparation from ECE312 should be sufficient for the course.

Tilted Classroom



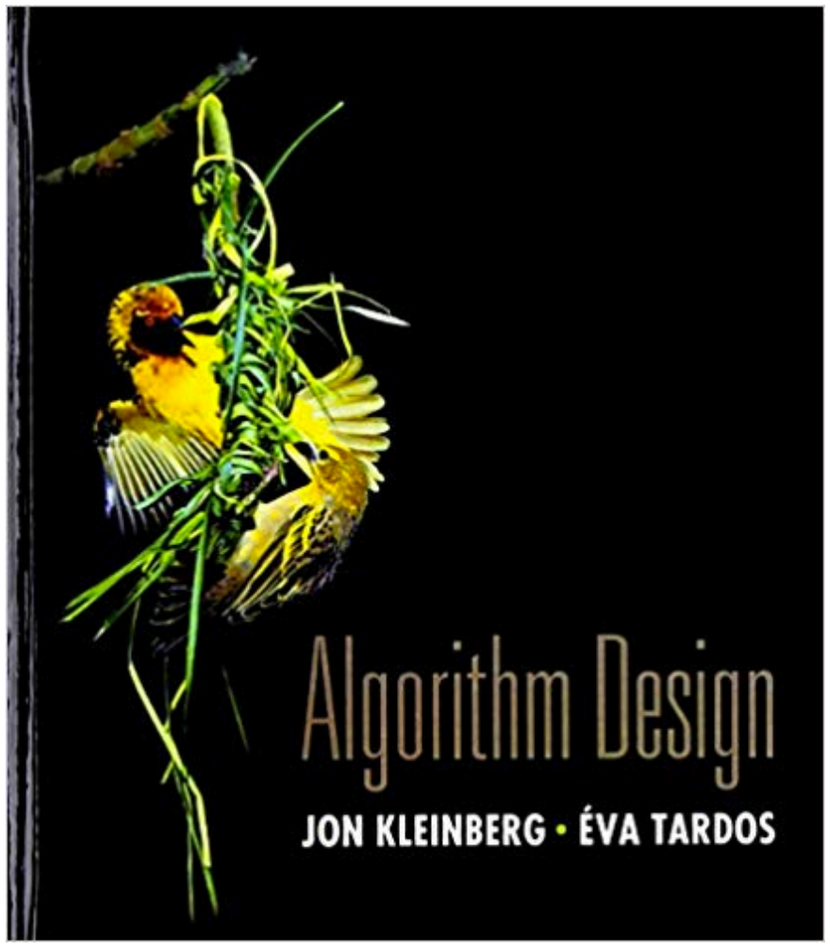
Lectures have two parts:

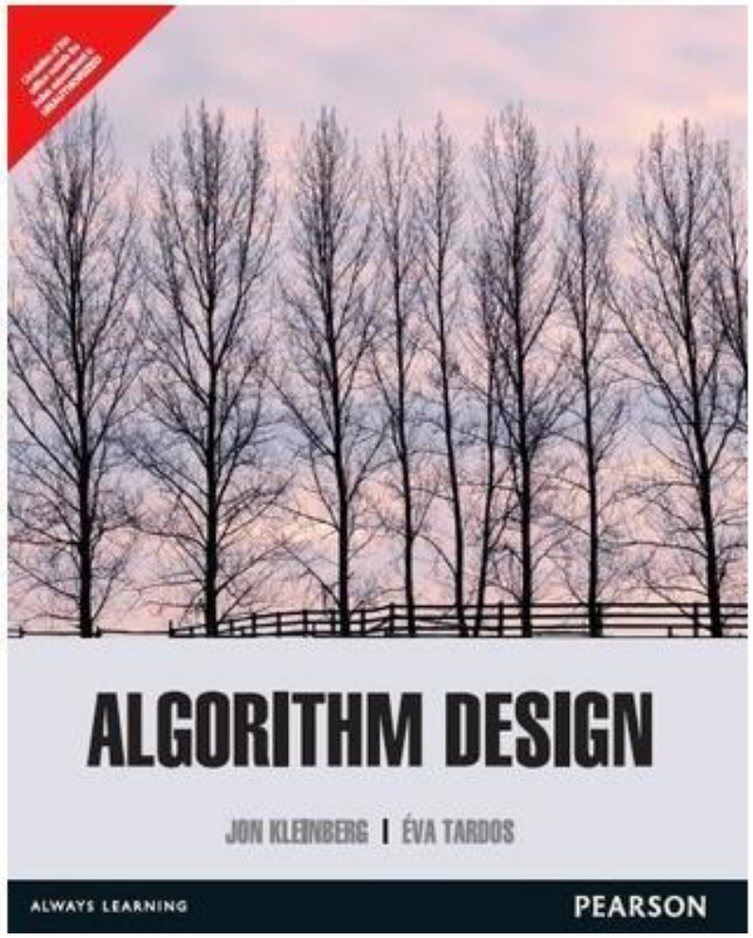
- **Pre-recorded Background** (~30-45 mins):
  - Watch on your own time
  - Submit Background Quiz (part of participation grade)
- **In-class Discussion**:
  - Discuss anonymized Background Quiz solutions
  - Additional content, answer questions, solve problems, motivate next lecture, etc.

Transition to my Office Hours immediately after class

Textbook

J. Kleinberg and E. Tardos. *Algorithm Design*. Addison Wesley, 2005.





International edition (cheaper)

Optional recommended texts

"CLRS"

- T. H. Cormen, C. E. Leiserson, R. H. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill, 2009 (Third Edition).

Testing

- 55% of final grade: 5 tests throughout the semester, lowest dropped
- 15% of final grade: final exam
- Tests during class time; final exam during exam period.

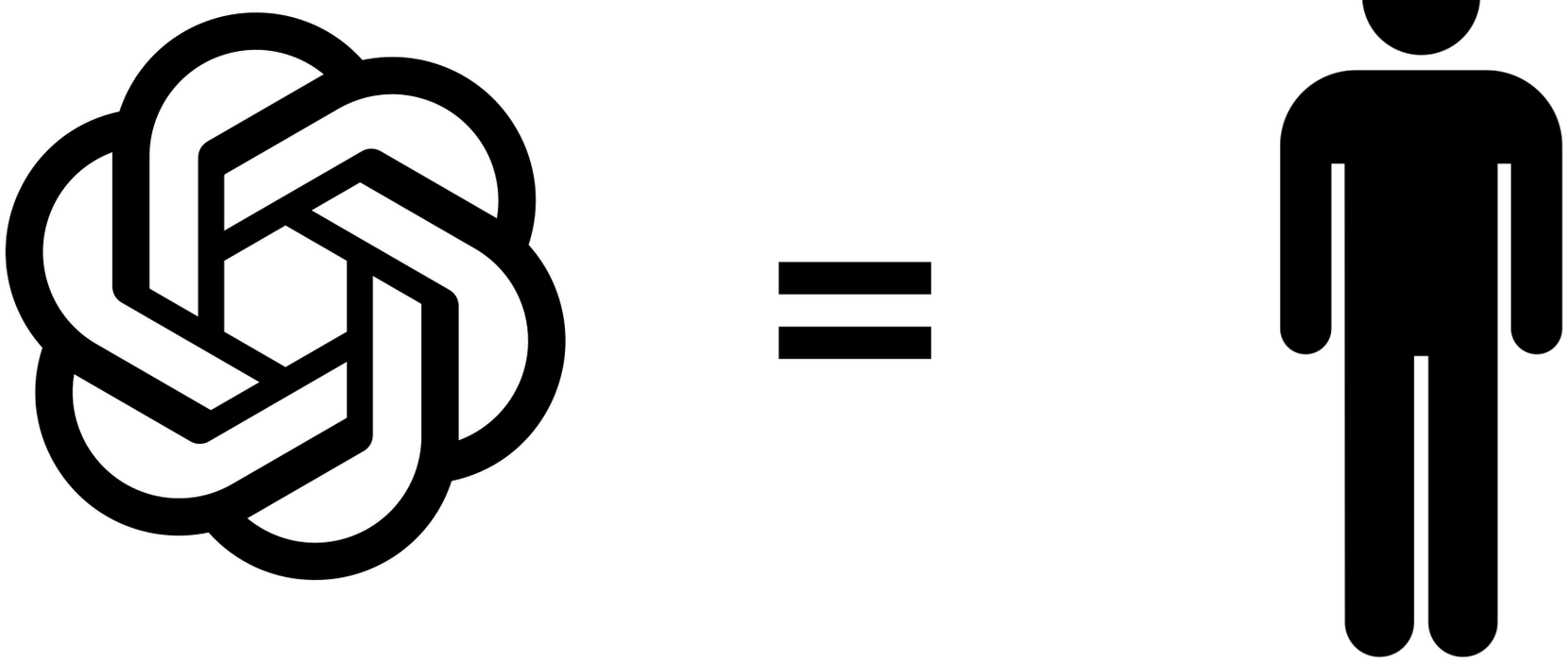
- Lots of practice problems and solutions throughout class, **but it's your responsibility to do them**
- Solutions will be made available for every test

Programming Assignments

- 20% of final grade: 3 programming assignments (each worth 6.67% of your final grade)
- Required to be in Java (starter code and grader in Java)
- Do not enable someone else to cheat (don't post code publicly)
- **You can discuss with others only at a conceptual level. Do not write or program while talking.**

You are encouraged to use books, your friends, the internet, ChatGPT, etc, to get solution ideas, but you may not copy/transcribe/transliterate code: get the idea, close the other resource, and then (**after enough time that the idea is in your long-term, not short-term, memory**) generate the code based on your own understanding. It is your responsibility to understand everything that you turn in. We reserve the right to ask you to explain any part of your homework assignment. If you are not able to explain what it means and why you chose it, that is presumed evidence of copying/cheating.

(This collaboration policy is adopted from Michael Ernst of University of Washington, and is used with permission)



For the purposes of the collaboration policy, ChatGPT and similar tools are considered people.

Participation

- 10% of final grade
- **Background Quizzes**
- Asking/answering questions in class or Ed Discussion

Participation: Background Quizzes

- One question for most pre-recorded Background Lectures. **Submit to Canvas prior to class (5pm!)**
- Graded for completion and effort
- 4 dropped to accommodate unexpected circumstances.
- **I will choose a few answers randomly in class for anonymous discussion**

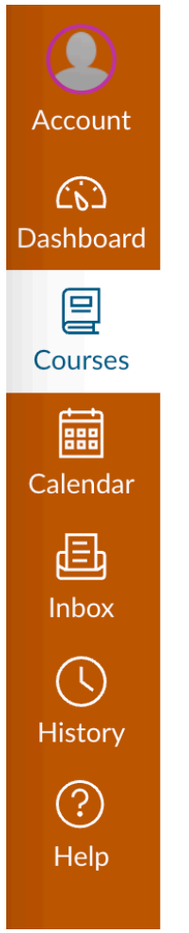
Grading Scale

Tests will be graded on a curve

The **final grades** will be assigned based on the standard numerical criteria:

A	93–100%
A–	90–92%
B+	87–89%
B	83–86%
B–	80–82%
C+	77–79%
C	73–76%
C–	70–72%
D+	67–69%
D	63–66%
D–	60–62%
F	0–59%

Tentative Course Plan



Fall 2023

Home

Modules

Assignments

Grades

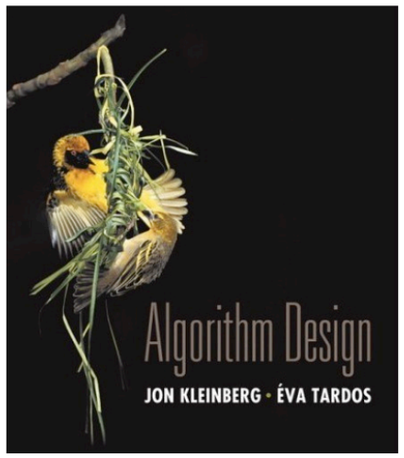
People

RegisterBlast

NameCoach

Ed Discussion

Fa23 - ALGORITHMS (17810)

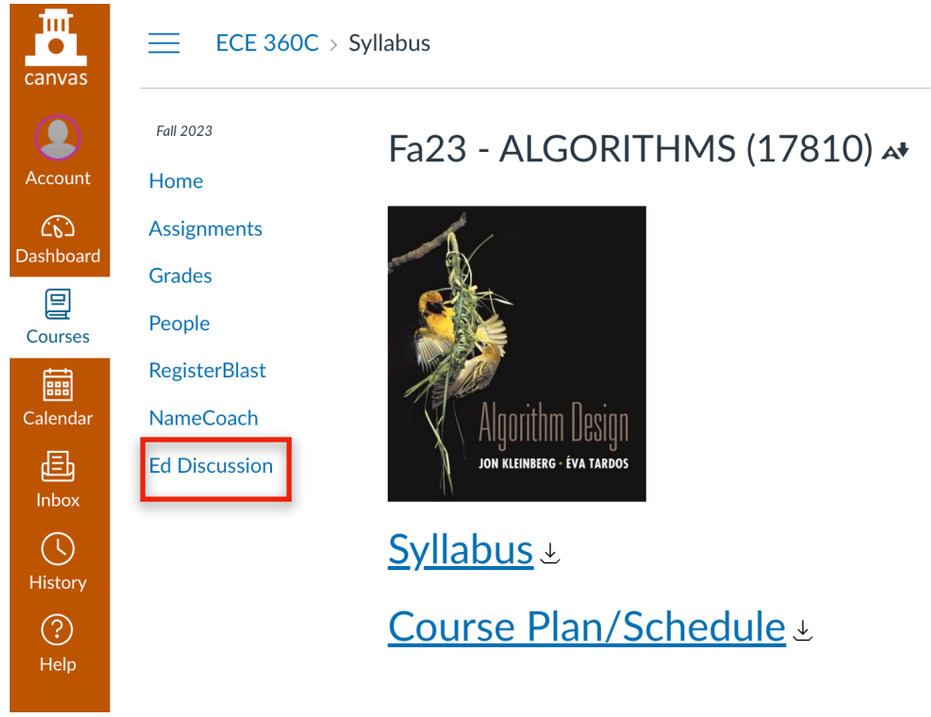


Syllabus, Slides: Course Logistics

Course Plan/Schedule

Ed Discussion

- The best way to ask me/TAs questions is through the discussion boards on **Ed Discussion**
- Other students can answer your question, or benefit from the answer



Teaching Assistants

graduate

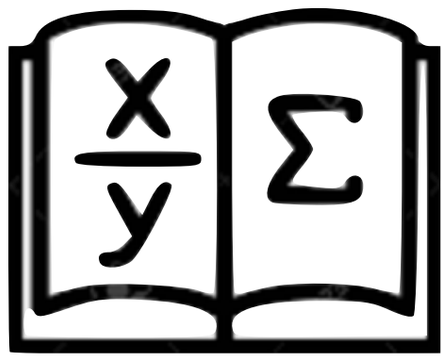
Alejandro R Gomez-Leos
alexgomezleos@utexas.edu

undergraduate

Jwalanthi Ranganathan
jwalanthi@utexas.edu

This is a **math class**

Programming assignments are used to help you understand the algorithms by implementing them



Common Misconceptions about the Class

1. If I memorize everything in the lectures (and the book), I will ace the exams!

2. I will skip the lectures, but study a ton of algorithms problems on the web, and I will ace the exams!

Common Misconceptions about the Class

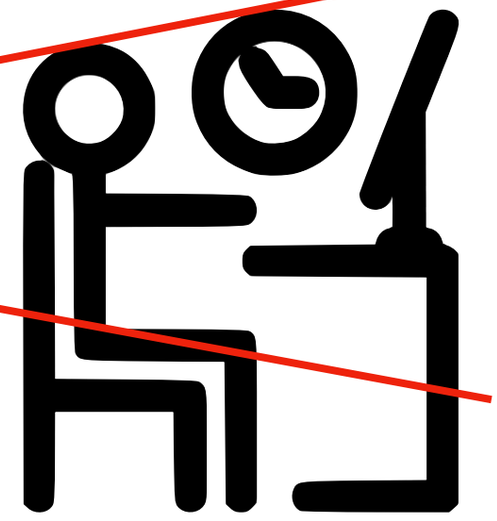
3. On exams, I have to write detailed pseudocode to describe my algorithms, making sure to not have off-by-one errors!

```
Algorithm: Prim-MST (adjMatrix)
Input: Adjacency matrix: adjMatrix[i][j] = weight of edge (i,j) (if nonzero)

   // inMST[i] = true once vertex i is in the MST.
1. Initialize inMST[i] = false for all i;
2. Initialize priority[i] = infinity for all i;
3. priority[0] = 0
4. numVerticesAdded = 0
   // Process vertices one by one. Note: priorities change as we proceed.
5. while numVerticesAdded < numVertices
     // Extract best vertex.
6.     v = vertex with lowest priority that is not in MST;
     // Place in MST.
7.     inMST[v] = true
8.     numVerticesAdded = numVerticesAdded + 1
     // Explore edges going out from v.
9.     for i=0 to numVertices-1
         // If there's an edge and it's not a self-loop.
10.        if i != v and adjMatrix[v][i] > 0
11.           if priority[i] > adjMatrix[v][i]
                // New priority.
12.              priority[i] = adjMatrix[v][i]
13.              predecessor[i] = v
14.           endif
15.        endif
16.    endfor
17. endwhile
18. treeMatrix = adjacency matrix representation of tree using predecessor array;
19. return treeMatrix

Output: Adjacency matrix representation of MST
```

Exception: programming assignments

Common Misconceptions about the Class

4. I impose on the professor's time if I attend office hours and ask questions. (The professor will deem my questions not worthy.)



5. The professors enjoy making the exams super-hard and watch the students suffer in agony. (It makes them feel powerful, and compensates for their inadequacies in other spheres of life.)