MACS 406: Design and Analysis of Algorithms (Jan 7)

Dinesh Mehta

- *Administrative Information:*

  1. *Meeting Time:* 10:00-10:50 am MWF.
  2. *Office Hours:* 11am-noon MWF or by appointment in Chauvenet 126.
  3. *Telephone*: (303) 273 3713.
  4. *Email*: dmehta@mines.edu.
  5. *Course Web Page*: (**Lecture Notes available here**)

     `http://www.mines.edu/~dmehta/Algorithms/`

- *Prerequisites:* MACS 213/MACS 223; MACS 262; MACS 358 (please do not take this in the same semester as 406). More specifically:

  1. *Basic data structures*: arrays, linked lists, queues, stacks, binary trees (including binary search trees).
  2. *Some familiarity with Asymptotic Notation*: big-Oh notation.
  3. *Sorting*: bubble sort, insertion sort, quick sort, merge sort.
  4. *Search*: Sequential Search, Binary Search.
  5. *Proofs*: by contradiction, induction.

- *Grading Procedures*:

  1. Five quizzes (100 pts *each*): will test your understanding of the theoretical aspects of the class. Total: 500 pts.
  2. Dynamic Programming Algorithms Project (group): 100 pts.
  3. LEDA assignment (individual): 100 pts.
  4. Final Programming Project (group): 200 pts.
  5. Homework Assignments (individual). These are problems from the text that will help you prepare for quizzes: 50 pts.
  6. Class Participation/Interaction: 50 pts. This part of the grade has been included to encourage students to be more interactive. The grade will be based on my (subjective) impression of your interest and enthusiasm. This will be measured by, among other things, your attendance, the number and quality of questions both during class and outside class, and your responses to my questions in class.
  7. How will group projects be graded?

     There will be two components to the grade for a group effort. I will assign a score based on the quality of the work. The group will assign effort percentages based on each member's contribution. Your individual grade will be a combination of both components. Example: A project is worth 50 points. I assign the project a score of 40 points. Suppose the group assigns effort percentages of 40%, 40%, and 20% to its three members A, B, and C, respectively. Then A and B will have a grade of $0.4 \times 40 \times 3 = 48$ points. C will receive $0.2 \times 40 \times 3 = 24$ points

8. Assignments and projects are due at the *beginning* of class on the assigned date. Late submissions will not be accepted, except in extenuating circumstances at the discretion of the instructor.
9. Examinations: Except for unforeseen emergencies, requests for make-up examinations will not be entertained. These requests should preferably be made before the examination and accompanied by supporting documentation.
10. Final Grades: Grades will be assigned based on your relative performance in the class; i.e., there are no hard cutoffs for each letter grade. However, 90% is typically a safe A, 80% a safe B, etc. *However*, in order to pass the class, a student must be deemed to have a passing total score **AND** a passing score in the programming component of the class.
11. Please see attached sheet that defines academic misconduct in the context of this class and sign the pledge.

- *Textbook*: Introduction to Algorithms, by Cormen, Leiserson, and Rivest. (*Second Edition*).
- *Supplementary Text*: The Algorithm Design Manual, by Skiena.
- *Course Calendar*

JANUARY

12 Introduction: (RA[1]: Skim Chapters 1 & 2
14 Asymptotic Complexity (RA: Chapter 3).
17 Recurrence Relations (RA: Section 4.1)
19 Recurrence Relations (RA: Section 4.2)
21 Recurrence Relations (RA: Section 4.3)
24 Heap Sort (RA: Sections 6.1 - 6.3)
26 Heap Sort, Quick Sort (RA: Sections 6.4, 6.5, 7.1)
28 Quiz 1 (Asymptotic Complexity, Recurrence Relations)
31 Lower Bounds, Radix Sort (RA: Sections 8.1, 8.3)

FEBRUARY

02 Medians and Order Statistics (RA: Sections 9.1)
04 Medians and Order Statistics (RA: Sections 9.3)
07 Hash Tables (RA: Skim Chapter 10, Read Sections 11.1, 11.2 upto Thm 11.1, 11.3 but not 11.3.3).
09 Binary Search Trees & Red Black Trees (RA: Sections 12.1, 12.2, 12.3, 13.1, 13.2)
11 Red Black Trees (RA: Section 13.3)
14 Dynamic Programming: Matrix Chain Product (RA: Section 15.2, 15.3).
16 Dynamic Programming: Matrix Chain Product (RA: Section 15.2, 15.3).
18 Quiz 2 (Sorting and Searching).
21 Dynamic Programming: Longest Common Subsequence (RA: Section 15.4).
23 Greedy Algorithms: Activity Selection (RA: 16.1)
25 Greedy Algorithms: Huffman Codes (RA: 16.3)
28 Greedy v/s Dynamic: Knapsack problem (RA: 16.2).

[1]Reading Assignment

MARCH
02 Graphs: Breadth First Search (RA: 22.1, 22.2)
04 Graphs: Depth First Search (RA: 22.3)
07 Catch-up class
09 Quiz 3 (Dynamic Programming & Greedy)
11 Graphs: Topological Sort (RA: 22.4)
14 Graphs: Minimum Spanning Trees (RA: Chapter 23)
16 Graphs: Shortest Paths (RA: 24.1, 24.2)
18 Graphs: Shortest Paths & Flow Networks (RA: 24.3, 26.1)
21 **Spring Break: no class**
23 **Spring Break: no class**
25 **Spring Break: no class**
28 Graphs: Flow Networks (RA: 26.2)
30 Graphs: Bipartite Matching (RA:26.3)

APRIL
01 Binomial Heaps (RA: Chapter 19)
04 Quiz 4 (Graphs)
06 Binomial Heaps and Amortized Analysis (RA: Chapter 19, Section 17.1)
08 Amortized Analysis (RA: Section 17.2, 17.3)
11 Fibonacci Heaps (RA: Chapter 20)
13 Fibonacci Heaps (RA: Chapter 20)
15 Fibonacci Heaps (RA: Chapter 20)
18 Disjoint Sets (RA: Chapter 21 except 21.4)
20 Disjoint Sets (RA: Chapter 21 except 21.4)
22 NP-Completeness
25 Quiz 5 (Advanced Data Structures)
27 NP-Completeness
29 NP-Completeness

MAY
02 NP-Completeness
04 NP-Completeness/Wrap-Up