CS 2223 Algorithms – D Term 2020

March 25, 2020

Instructor:

Michael Engling mengling@wpi.edu

Lectures:

MTΘF 3:00–3:50
Zoom Room: `https://wpi.zoom.us/j/6982696493`

Office Hours:

MT RF 4–4:30pm (Immediately after class. . . in the Zoom Room)
Wednesday 3–4:30pm (Classtime on non-class day. . . in the Zoom Room)
Also by arrangement

Teaching Assistants / Office Hours:

Tariq Rakha TBD
Zekun Dai TBD

Recommended Background:

- CS 2022 $\wedge$ CS 2102
- Discrete Math and Java.

Course Description:

Building on a fundamental knowledge of data structures, data abstraction techniques, and mathematical tools, a number of examples of algorithm design and analysis, worst case and average case, will be developed. Topics include greedy algorithms, divide-and-conquer, dynamic programming, heuristics, and probabilistic algorithms. Problems will be drawn from areas such as sorting, graph theory, and string processing. The influence of the computational model on algorithm design will be discussed. Students will be expected to perform analysis on a variety of algorithms.

Recommended Textbooks:

Introduction to the Design and Analysis of Algorithms, 3rd edition, 2012;
by Anany Levitin; ISBN-10: 0-13-231681-1; ISBN-13: 978-0-13-231681-1

`https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Introduction%20to%20the%20Design%20and%20Analysis%20of%20Algorithms%20%283rd%20ed.%29%20%5BLevitin%202011-10-09%5D.pdf`

Other Textbooks:

Introduction to Algorithms, 3rd edition, MIT Press, 2009;
by T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein;
ISBN 978-0-262-03384-8
`URL:https://mitpress.mit.edu/books/introduction-algorithms`

Algorithm Design, Addison-Wesley, 2005;
by J. Kleinberg, E. Tardos;
ISBN-10 0-321-29535-8; ISBN-13: 978-0-321-29535-4
`https://www.cs.princeton.edu/~wayne/kleinberg-tardos/`

Algorithms in a Nutshell: A Practical Guide, O'Reilly, 2nd edition, 2016;
by by George T. Heineman, Gary Pollice, Stanley Selkow ;
ISBN-10: 1-491-94892-2 ; ISBN-13: 978-1-491-94892-7
`http://www.r-5.org/files/books/computers/algo-list/common/Heineman_Pollice_Selkow-Algorithms_in_a_Nutshell-EN.pdf`

Student Accomodations:

Students with disabilities who need to utilize accommodations in this course are encouraged to contact the Office of Disability Services (ODS) as soon as possible to ensure that such accommodations are implemented in a timely fashion. That office can be contacted via email: DisabilityServices@wpi.edu, via phone: (508) 831-4908, or in person: 124 Daniels Hall. If you have approved accommodations, please request your accommodation letters online through the Office of Disability Services Student Portal.

Objectives:

- Calculate the asymptotic running time of standard algorithms.
- Explain the meaning of big Oh, Theta, and Omega notations and use them to reason about the performance of diverse algorithms.
- Use the Master Theorem to prove asymptotic assumptions.
- Implement standard algorithms using graphs and weighted graphs in an object-oriented programming language, e.g. Depth-First Search, Breadth-First Search, Minimum Spanning Tree algorithms.
- Compare and analyze basic and advanced sorting algorithms.
- Implement advanced search trees such as B-trees, AVL trees, & 2-3 trees.
- Implement numerical algorithms such as Gaussian elimination, binary exponentiation, and the simplex method.
- Apply standard algorithm design techniques such as the greedy technique, dynamic programming, hashing, space/time trade-offs, reduction, backtracking, and branch-and-bound.

Grading: (Total 1000 points)

Programming Assignments (Homeworks) **[500 points]**

There will be six programming assignments using Java. We will begin with an introductory exercise worth 50 points. Homeworks 2 and 3 will be somewhat more involved and be worth 75 points each. The final three Homeworks will each have a more sophisticated implementation and count for 100 points each.

Midterm Exam **[140 points]**

Final Exam **[150 points]**

Quizzes (Best 21 of at least 24) **[210 points]**

Final letter grades will be assigned on the following basis:

A = 900 points
B = 800 points
C = 655 points
NR < 650 points

Programming Assignments / Homeworks:

Homework 0	Introduction and Setup	10 (Bonus) Points	Due March 28
Homework 1	Double Trouble	50 Points	Due April 4
Homework 2	TBA	75 Points	Due April 11
Homework 3	TBA	75 Points	Due April 18
Homework 4	TBA	100 Points	Due April 25
Homework 5	TBA	100 Points	Due May 2
Homework 6	TBA	100 Points	Due May 9
TOTAL		500 Points	

Homeworks will be assigned on Wednesdays. They will be due at 2pm on the second Saturday after they are assigned, so you will always have more than 9 full days to work on them and ask questions about them. Homeworks will usually contain at least two significant problems from topics recently discussed in class sessions. Do not put off working on them until the last day or two.

Late Homework Policy:

- Homeworks submitted after 2 pm on the Saturday due date or anytime on the following Sunday will incur a 10% penalty.

- Homeworks submitted on Monday will incur a 25% penalty.

- Homeworks not submitted by midnight Monday will receive a grade of 0.

Academic Integrity:

We will, perforce, be conducting all our interactions online, from a distance. We will make use of the Piazza forum within Canvas, and we'll have numerous opportunities to collaborate and share ideas, dispersed though we may be. That being said, it is incumbent on you to do your own work and submit your own code. You are encouraged to consult me, the TAs, and your classmates. Annotate your work in comments; credit those who spark your understanding; write your own code for the Homeworks, and do the Quizzes and Exams solo.

Week	M	T	W	Θ	F
0	Data Structures Non-Recursive Algorithms O, Θ, Ω notations	March 24	March 25 Introduction Terminology Day 0 **HW 0 Due: Sat 2pm**	March 26 Size of Input Orders of Growth Day 1	March 27 Recurrence Relations Day 2
1	March 30 Selection Sort Bubble Sort Day 3	March 31 Searching Depth First Breadth First Day 4	Brute Force Exhaustive Search Decrease & Conquer **HW 1 Due: Sat 2pm**	April 2 Insertion Sort Topological Sorting Day 5	April 3 Generating Permutations and Sets Day 6
2	April 6 Mergesort Quicksort Day 7	April 7 Heapsort Day 8	Master Theorem Divide & Conquer Transform & Conquer **HW 2 Due: Sat 2pm**	April 9 Master Theorem Day 9	April 10 Horner's Rule Square & Multiply Day 10
3	April 13 DP Basics Day 11	April 14 Knapsack Problem Day 12	Dynamic Programming Transform & Conquer **HW 3 Due: Sat 2pm**	April 16 Gaussian Elimination Day 13	April 17 **Midterm Exam** Day 14
4	April 20 Patriot's Day No Class	April 21 Minimum Spanning Trees Day 15	Greedy Algorithms **HW 4 Due: Sat 2pm**	April 23 Dijkstra's Algorithm Day 16	April 24 Huffman Codes Day 17
5	April 27 Closed & Open Hashing Day 18	April 28 Backtracking Day 19	Space and Time Trade-Offs **HW 5 Due: Sat 2pm**	April 30 Horspool's Algorithm Day 20	May 1 Bipartite Matching Stable Marriage Day 21
6	May 4 Branch and Bound Day 22	May 5 Max Flow and Min Cut Day 23	Iterative Improvement Limitations of Algorithmic Power **HW 6 Due: Sat 2pm**	May 7 Counting Paths in Graphs Day 24	May 8 AVL Trees Day 25
7	May 11 P, NP, NP-complete Day 26	May 12 **Final Exam** Day 27	P, NP, NP-complete Final Exam	May 14	May 15