CSCI 6632/3326 — Algorithms — 13 Fall

Course Objectives

By the end of this course you should know:

- Data structures like balanced binary search trees, graphs and disjoint set union-find representations.
- How to analyze an algorithm to see if it is efficient.
- Basic algorithmic design techniques like greedy algorithms, divide and conquer, dynamic programming.
- Basic sorting algorithms.
- Basic graph algorithms.
- A few more advanced algorithms in areas like number theoretic algorithms for public key cryptography, string matching algorithms, data compression algorithms.
- NP-Completeness: why certain problems are considered intractable.

Instructor contact information

Barun Chandra, 932-7089, Buckman 244, bcdefend-1@yahoo.com This is the email address you should use for this course and for any questions you might have for me as graduate coordinator. If you need to send me an "official email" which might need to be forwarded to other faculty or staff at UNH or need to give my email address to a prospective employer etc, please use my official UNH email address: bchandra@newhaven.edu

Office Hours

To be announced.

Prerequisites

Data Structures.

Books

These are available in the library, some are on reserve and some can be checked out.

Text book: Introduction to Algorithms, Cormen, Leiserson, Rivest, M.I.T. Press. Third Edition. ISBN-10: 0262033844 ISBN-13: 978-0262033848 There are multiple copies of the2nd edition textbook in the library, one is on reserve, the others can be checked out. There is also an earlier first edition which can be checked out. If you have the first edition, you will probably be able to manage, but you might need to photocopy a few pages from the second edition, and the chapter numbers and some of the problem numbers will be different.

Reference book: Fundamentals of Algorithms, Brassard and Bratley, Prentice-Hall. On one week reserve. This covers a lot of the topics also covered in the text book, and a number of topics covered in the course are from this book.

Reference book: The Algorithm Design Manual, Steven Skiena, Springer-Verlag.
On one week reserve. This has a lot of interesting case studies and applications of algorithms and techniques.

Reference book: Computer Algorithms, Baase and Van Gelder, Addison Wesley. Can be checked out.

If you have a hard time following the textbook, you might want to consider getting Brassard and Bratley; it covers a number of the same topics.

Course Requirements

4 written homeworks, 4 quizzes, final. There will be no midterm. There will not be much required programming for the class, but a lot of the extra credit problems will be programming problems. The quizzes and the final will be open book, open notes.

Grades

Homeworks(35%), quizzes(30%), final(35%). Your lowest homework score and your lowest quiz score will not count towards your grade.

Extra Credit Problems

I will assign a number of extra credit problems; you should turn these in the week of the final. Unlike the required problems, you will get no partial credit for the extra credit problems i.e. either it is completely correct or it is wrong, so there is no point in handing in a weak or incomplete submission. In terms of your course grade, the extra credit problems have a fairly small impact: I only look at how many extra credit problems you have done if you are right on the borderline of two grades, and if you have done some your grade can go up. Hence the required problems should get a much higher priority then the extra credit problems. There will be quite a lot of extra credit problems, so you should pick only those which look interesting to you.

Requirements for a grade of A+

To get an A+, you have to do the following:

1. You must do extremely well on the required work.
2. You must do at least 3 extra credit problems correctly.

Late and Incomplete Policy

All work is due at the beginning of class on the due date. The rule for late assignments is as follows:

- Assignments handed up to a week late: late penalty of 10%.
- Assignments handed more than a week but less than two weeks late: late penalty of 50%.
- Assignments handed more than two weeks late will not be accepted.

If you have completed some problems and not others by the due date, please turn in the problems you have finished. You will not be assessed the late penalty on the problems you turn in on time. No Incompletes will be given for the course. These policies will be strictly enforced.

A Friendly Reminder about Plagiarism

Sharing ideas about assignments is acceptable, even encouraged. The writing of the assignments or programs, however, should be strictly an individual affair. The following will all be considered as cases of plagiarism: 1. One student does the work and another copies the work. 2. Two students write an assignment together. 3. A student gets some material from a book or the internet and does not say where it came from. 4. A student turns in a program and output, and the output is not from that program. If you have any questions about what is an acceptable level of collaboration, please ask me. Any case of plagiarism will result in *failing the course for any student who is involved in it.*
Some of you may be coming from institutions where plagiarism is not taken seriously. It is taken seriously at UNH, and certainly taken seriously by me. I enforce the plagiarism policy strictly. If you are ever thinking about cheating, please keep one thing in mind: in the past, in classes I have taught, more people have recieved a failing grade because of plagiarism than for any other reason.

Appropriate and professional behavior in class

- Cell phones: either switch off your cell or put it on silent mode, and if you need to answer a call, please leave the classroom.
- Don't text or read email or surf the web during class.
- Please try to be on time; students who are regulary late for class disturb their fellow students and leave a strongly negative impression on the instructor.

Powerpoint presentations

Many of the lectures will use powerpoint presentations. I will email you the presentation before the lecture. So the first thing I need from you is your email address: please send me an email saying which course you are taking so I will find out your email address.

Blackboard

The powerpoint lectures will also be put on blackboard; you should familiarize yourself with blackboard.

HW guidelines for the paper and pencil problems

For the "paper and pencil" problems, you have to submit hard (paper) copies - **email submissions will not be accepted**.

You will loose points if you don't follow these guidelines:

- All work is to be done using a word processor i.e. your homework should not be handwritten. The exception to this is if there are complex formulae or complex figures; it is OK to do these by hand.
- All work has to be submitted in the correct order (i.e. problem 1 before problem 2 before problem 3 ...) and should be neatly stapled.
- You don't need to copy the question on your answer sheet; you only have to write the answer.
- Please read the question carefully and answer only what is being asked for, no more and no less. As an example, supposed you are asked to give a brief answer as to the difference between a stack and a queue. What your answer should say is that a stack is LIFO and a queue is FIFO. Here are some examples of what **you should not do:**
  - Start quoting from the textbook as to what stacks are and what queues are.
  - Start explaining why stacks are useful and where they are used.
  - Give a long explanation as to why data structures are wonderful.

  If you do any of the above, and include a lot of material which is not directly relevant to what the question is asking, you will loose points.
- When a question asks for an answer and says "no explanation neccessary," if you need to do some calculations etc, do them on a separate sheet of paper. For example, if I tell you that a queue has the elements $5, 3, 8, 2, 6$, and I ask you to tell me what will be the third element dequeued (no explanation neccessary), all I expect to see on your answer sheet is 8, *and nothing else.*
- If you are getting information from a book or a web site (see the policy on plagiarism) you should attribute it, understand what is written, and then write the answer in **your own words.**

Tentative Syllabus

This is the order in which I plan on covering the following topics. BB refers to the reference book Brassard and Bratley.

Week 1 Review of elementary math concepts: logarithmic and exponential functions, Σ, Π. Alorithms: basic concepts: Algorithms: how to specify, correctness, efficiency, worst case, average case and best case analysis, order of growth, big-oh $O()$ notation – Chapters 1 and 3. Sorting: Insertion Sort – 2.1. Merge Sort – 2.3.

Week 2 Sorting: $\Omega()$ and $\Theta()$ notation – 3.1. Counting Sort – 8.2. Lower Bounds for Comparison Based Searching and Sorting Methods – 8.1.Review of Data Structures: Linked Lists, Stacks, Queues – 10.1,10.2. Trees – Appendix B.5 and 10.4. Binary Search Trees and Tree Traversals – 12.1, 12.2, 12.3. Heaps, Priority Queues, Heap Sort: Chapter 6.

Weeks 3 Review of elementary math concepts: permutations and combinations, proof by induction, proof by contradiction. Elementary Graph Algorithms: Graphs – Appendix B.4. Graph Representations – 22.1. Depth First Search – BB. Topological Sort – BB. Breadth First Search (description but not the algorithm) – BB.

Weeks 4 Greedy Algorithms: Introduction – 16.2. Job Scheduling – BB. Minimum Spanning Tree Algorithms: Connected components. Kruskal's algorithm. Prim's algorithm (description but not the algorithm) – Chapter 23.

Weeks 5 Greedy Algorithms: Djikstra's Single Source Shortest Path Algorithm – 24.3. Divide and Conquer: Introduction – 2.3. Recurrence Relations, Master Theorem – 4.3. Multiplication of Large Integers – BB.

Week 6 Divide and Conquer: Quicksort: Worst Case, Average Case, Randomized Quicksort – Chapter 7. Public Key Cryptograph Modular Arithmetic – 3.2. RSA – 31.7. Fast Modular Exponentiation – BB.

Week 7 Text Compression: Huffman Codes – 16.3. Balanced Binary Search Trees: Red-Black Trees – Chapter 13.

Week 8 Text Compression: Ziv-Lempel zip compression algorithm – Stallings pages 391-393. B-trees: Introduction – 19.1. Search and Insertion – 19.2.

Week 9 NP-Completeness: Introduction, Optimization Vs Decision Problems, Reduction from Clique to Independent Set, Reduction from Hamiltonian Cycle to TSP – Chapter 34. NP, NP-hard, and NP-complete problems, Reduction from SAT to CLIQUE – Chapter 34.

Week 10 Amortized Analysis: Stack operations, – 17.1, Union-Find on Disjoint Sets with Linked Lists – 21.1, 21.2. Union-Find on Disjoint Setswith Tarjan's Forest representation– 21.3. String Matching: Overview – 32.1. Rabin-Karp Algorithm for String Matching – 32.2.

Week 11 Dynamic programming: Introduction and Fibonacci Numbers example – 15.3. Floyd's Algorithm for All Pairs Shortest Paths – 25.2.

Approximation Algorithms for NP-hard problem: Vertex Cover – 35.1,

Week 12 Approximation Algorithms for NP-hard problem: TSP – 35.2. Probabilistic Primality Testing: Miller-Rabin – 31.8. Advanced Topics: Time permitting, we will also cover some of the following: Extended Euclidean Algorithm for RSA – 31.2. Approximate String Matching with Dynamic Programming - Skiena pages:60–62,406–409. Boyer-Moore Algorithm for String Matching – 34.5 from CLR 1st edition.

Week 13 Final.

Tentative Homework and Quiz Schedule

- Homework 1 - handed out on week 1, due on week 3, will cover material from weeks 1, 2.
- Quiz 1 - on week 4, will cover same material as HW 1.
- Homework 2 - handed out on week 4, due on week 6, will cover material from weeks 3, 4, 5.
- Quiz 2 - on week 7, will cover same material as HW 2.
- Homework 3 - handed out on week 7, due on week 9, will cover material from weeks 6, 7, 8.
- Quiz 3 - on week 10, will cover same material as HW 3.
- Homework 4 - handed out on week 9, due on week 11, will cover material from weeks 9, 10, 11.
- Quiz 4 - on week 12, will cover same material as HW 4.
- Extra Credit Problems - due on week 13.