Course Information

Welcome to Data Structures. This course will introduce students to a wide array of data structures, mechanisms of analysis, along with when and how to use them. Students will:

1. Learn about the wide variety of data structures available and how they can completely change how we approach problems, with the aim of being able to recognize when and how to apply this knowledge.
2. Design and analyze nontrivial data structures and see related algorithms.
3. Solve recurrences and structure proofs.
4. Code. Code a lot. Like, really absurd amounts of coding.

We will cover basic data structures (linear lists, arrays, stacks, queues), analysis, trees, hash-tables and related constructions, skiplists, self-organizing lists, graphs, strings, integers, geometry, and structures optimized for the memory hierarchy; possibly some succinct structures. Abstract data types. Primitive data types; some background on word RAM. Elementary principles of software engineering, as well as some algorithms intrinsically related to the data structures taught in class. A more detailed syllabus is provided later in this document.

This handout describes basic course information and policies. Most of the sections will be useful throughout the course. The main items to pay attention to **NOW** are:

- ◇ Make sure you are signed up properly on AMS and Piazza.
- ◇ Add this course on gradescope using the code 6GZKVG to submit written assignments. Instructions for coding assignments will be included within those documents.
- ◇ Please note, and carefully adhere to, the collaboration policy.

1 Course Management

We will be using Gradescope to submit and grade assignments, and Piazza to post course material, answer questions, and for announcements, such as changes to office hours. Coding assignments might be handled differently; you will receive detailed instructions on the same later.

2 Staff

The lecturer for this course is Debayan Gupta (`debayan.gupta@ashoka.edu.in`). Please call me Debayan; if you feel uncomfortable with that, I also answer to Prof. Gupta, though I prefer the former.

Our TAs (in order of first names):
Bhumika Mittal
Diya Roongta
Jivansh Sharma
Kahaan Shah
Karnav Popat
Shivam Kedia

3 Prerequisites

Discrete Mathematics, Introduction to Computer Programming.

4 Lectures

Lectures will be held in AC03 LR 005, 11:50am-1:20pm on Tuesdays and Thursdays. You are responsible for all material presented in lectures, including oral comments made by the lecturer. All lectures will be recorded.

4.1 Syllabus

This course acts as a necessary ramp to Algorithms, and must synergize well with Discrete Mathematics (and Probability and Statistics). It is critical that students understand the difference between abstract data types and implemented data structures, as well as proofs of completeness and correctness. Students are expected to be able to produce mid-size

programs (say, 2 kloc). Lecture 17 onwards, we deal with advanced material: it is likely this will have to be extended (or reduced) depending upon how the class is doing.

1. Algorithmic Thinking; Arrays and Lists. Table doubling. **1.5 hours**
2. Asymptotics and a crash course in complexity. This may include amortized and competetive analysis (mostly readings). **0.5 hours**
3. Insertion Sort, Selection Sort, Merge Sort, Sorting Lower Bounds, Counting Sort, Radix Sort. (Basic coverage in class, mostly readings) **1.5 hours**
4. Stacks and Queues. Priority queues, and connections to sorting.
   Problem in focus (next few lectures also): Huffman encoding. **2.5 hours**
5. Heaps, Fibonacci heaps. Heapsort and priority queues continued. **1.5 hours**
6. Binary Search Trees. BST Sort, dynamic optimality. **1.5 hours**
7. AVL Trees, AVL Sort. **1.5 hours**
8. B and B+ Trees. Usage in databases, range queries.
   Optionally Zip, red-black, fusion, and splay trees. **3 hours**
9. Direct access array, Hashing, including both SUHA and Universal hashing.
   Optionally, perfect and cuckoo hashing. **1.5 hours**
10. Bloom filters and Count-min sketches. Problem in focus: sublinear existence and counting. **1.5 hours**
11. Skiplists. Randomization, tree-alternative. **1.5 hours**
12. Quadtrees
    Problem in focus: k-nearest neighbours, fast multipole method. Geometry of a given circle and rectangle overlapping. **1.5 hours**
13. Breadth-First Search (BFS), Depth-First Search (DFS), Topological Sort.
    Connections to shortest paths, maze exit. **1.5 hours**
14. Single-Source Shortest Paths Problem, Relaxation, Dijkstra. **1.5 hours**
15. Bellman-Ford, dynamic graphs.
    Optionally, dynamic connectivity with Holm forests. **1.5 hours**
16. Union Find, self-organizing lists. Recall competitive analysis background.
    Problem in focus: connectivity, percolation. **1.5 hours**

17. Optimizing for the memory hierarchy: cache-oblivious B-trees, file structures and file maintenance, list labelling, cache oblivious priority queues.
    Optionally, distribution sweeping (funnelsort). Segue to geometry: cache-oblivious orthogonal 2D range searching. **3 hours**

18. Geometry: Full discussion on range trees. Dynamizing augmentation via weight balance, fractional cascading, 3D orthogonal range searching. **3 hours**

19. Strings. Searching static strings. Range minimum queries.
    Focus on computational biology examples. Suffix trees and arrays. Connect problems: Least common ancestor, k-th ancestor in constant time vs minimum number in a given interval of a pre-processed array **3 hours**

20. Integers. Predecessor lower bounds using van Emde Boas trees. Proof using round elimination.
    Problem in focus: Sorting on word RAM and links to priority queues. **3 hours**

21. If time permits, some introductory discussions on persistent and retroactive structures.

4.2 Office Hours

There will be lots of office hours pretty much every day of the week. We strongly recommend that you make efficient use of your and your TAs' time by sending your questions to the TAs at least a few hours beforehand.

5 Problem Sets

Problem sets (aka homeworks/assignments) will be assigned throughout the semester. Overall, we shall probably have 5-7 problem sets. The due date will always be written on the problem set itself. Homework must be turned in by 2:00 P.M. on the due date.

- **Late submissions:** We will not usually allow late submissions unless there are major extenuating circumstances, such as a medical issue.

  The only way to submit late is to email your solution to your TA and cc the exception-handling czar (you'll find out what this is later). We will sometimes allow small (< 30 min) delays, but we intend to keep an eye out for habitual offenders and may reject such submissions arbitrarily.

- **Discount Policy:** Your lowest homework score will be ignored.

- **Office Hours:** There will be no office hours on the day a problem set is due.
- **Submission Format:** Solutions to written parts of the problem set should be submitted online to gradescope in a single PDF file. If the file does not clearly indicate which parts the solutions refer to, or has parts missing, it is assumed that the student did not attempt that part of the problem. Therefore, before submitting, make sure all of your work is included in the PDF file.

  Start each question on a new page and mark the top of the page with the following: (1) your name, (2) the question number, and (3) the names of any people you worked with on the problem (see Section 10), or "Collaborators: none" if you solved the problem entirely by yourself.

  The problem sets may include exercises that should be solved but not handed in. These questions will be clearly marked and are intended to help you master the course material. Material covered in exercises will be tested on exams.
- **Regrade Requests:** Any student who feels that a problem set was not graded properly may submit a regrade request through Gradescope within one week of the graded assignment being returned to the student. Please note the following before submitting a regrade request:

  1. You should carefully read the posted solutions for the problem in question.
  2. Indicate which rubric items you deserve (if applicable), where in your solution write-up you address them, and explain why you deserve extra points. Any regrades without justification will not be processed.
  3. The course staff reserves the right to regrade the entire assignment, and your grade may increase or decrease as a result of a regrade.
  4. **Important:** Lots of requests from the same person (hoping to somehow get extra points) and nonsensical requests will be dealt with harshly. This sort of thing wastes the time of the course staff and means that we can't help other students who might actually need it.

  If you are still unsatisfied with your grade after the regrade, please email Debayan.

6 On the Importance of Clarity

You should be as clear and precise as possible in your write-up of solutions. Understandability of your answer is as desirable as correctness, because communication of technical material is an important skill.

A simple, direct analysis is worth more points than a convoluted one, both because it is simpler and less prone to error, and because it is easier to read and understand. Sloppy

answers will receive fewer points, even if they are correct, so make sure that your solutions are concise and well thought-out.

Sometimes, you will be asked to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode (if you don't know what that is, we'll go over it in class!).
2. A proof (or indication) of the correctness of the algorithm.
3. An analysis of the asymptotic running time behavior of the algorithm (again, we'll learn more about this later).
4. Optionally, you may find it useful to include a worked example or diagram to show more precisely how your algorithm works.

Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted or obtuse descriptions.

7 Exams

There will be two exams in the course. These will be 1.5 hours long; the second exam will be cumulative. There will be no final exam during exam week in December; instead, we will test your understanding of the concepts learned in class through the puzzle week event (see the next section).

More details about your exams will be released closer to the date. (We'll have lots of tiny features to minimise cheating.)

Attendance at the exams is mandatory. Legitimate conflicts can be discussed with the teaching staff but must be due to extenuating circumstances and discussed in advance. If a student misses either exam due to an emergency, make-up exams may be offered at the discretion of the instructor.

Regrade requests. Any student who feels that a quiz or final exam was not graded properly may submit a regrade request. The request must be made online (via gradescope) by the announced deadline. The request should include a detailed explanation of why she or he believes that a regrade is warranted. Please make sure you read the solutions carefully before requesting a regrade.

8 Puzzle Week

At the end of the semester, we will host a week-long coding and puzzle-solving scavenger hunt. You will hear more about this later in the semester. Attendance is mandatory and we cannot offer a make-up for this event.

9 Grading Policy

The final grade will be calculated as follows:

Homework	25%
Exam 1	25%
Exam 2	25%
Puzzle Week	25%

10 Collaboration Policy

We encourage you to collaborate with your peers to deepen your understanding of the course material. However, you should approach collaboration *on problem sets only* with care, and follow the guidelines below. Copying **without understanding** from online resources, books, or notes from previous versions of this or other classes is strictly forbidden — such mindless copying will be considered a serious offense and dealt with accordingly. Using online resources for information is fine, as long as you process, understand, and cite them properly.

1. **You should spend at least 30–45 minutes trying to solve each problem entirely by yourself.** If you find yourself unable to progress, you can seek help, either by approaching the TAs, or by using the forum, or by collaborating with your peers.
2. **Do not be a Spoiler.** If you already solved the problem, do not give away the answer to your friend. The best way you can help your friend is to give hints and allow her or him the pleasure of coming up with the answer her/himself. Our past experience has overwhelmingly shown that students who do not attempt the problem sets on their own generally perform poorly on the exams, and thus in the class overall.
3. **You must write up each problem solution entirely by yourself without assistance,** even if you collaborate with others or look up stuff online to solve the problem. Doing otherwise will be considered plagiarism, an academic offense with serious repercussions. You are asked on problem sets to identify your collaborators. If you did not work with anyone, you should write "Collaborators: none."

> **It is a serious violation of this policy to submit a problem solution that you cannot orally explain to a member of the course staff.** Plagiarism and other dishonest behavior cannot be tolerated in any academic environment that prides itself on individual accomplishment.

If you have any questions about the collaboration policy, or if you feel that you may have violated the policy, please talk to one of the course staff. Although the course staff is obligated to deal with cheating appropriately, we are far more understanding and lenient if we find out from the transgressor himself or herself rather than from a third party or on our own.

Needless to say, **no collaboration whatsoever is permitted during exams**.

11 Textbook

We will not use any textbook. The internet is your friend.

Course notes will be posted every week. In addition to this, there will be readings from various books, but the necessary excerpts will be provided. If you're interested and want to read the rest of some of these books, first check the library, then ask the professor to help you find a copy.

12 Advice and resources for effective learning

Because of the conceptual nature of the material, just attending lectures and doing the homework are unlikely to be sufficient for learning all the concepts. **Setting aside time to do the reading and to study your notes from lecture and recitation is generally necessary to truly learn and internalize the material**, and to be able to apply it in new ways later in the course as well as for the rest of your life.

Homework is essential for learning the material. Rather than thinking of problem sets as just a requirement, recognize them as an excellent means for learning the material, and for building upon it. Spread out the time you have to work the problems. Many people learn best by reading the problems long before they are due, and working on them over the course of a whole week; they find that their minds make progress working the problems in the background or during downtime throughout the day. Few people do their best learning the night before an assignment is due. Work with others if that is helpful, but with the goal of learning first and solving the problems second. **It is worth reading the posted homework solutions, even if you received full credit**. Often the clarity of explanation or details of implementation are different from the way you were thinking about things in ways that can improve your learning.

Don't hesitate to ask for help. This class is largely conceptual, and the concepts tend to build on one another. **If you are having trouble understanding the material, it is important to catch up rather than risk falling further behind**. We can help.

Office hours are a particularly useful mechanism for learning material and working through difficulties on problem set assignments. Moreover, if you have questions about the course or problem sets, please use the forum as opposed to emailing an individual TA or lecturer—that will give you a better chance of getting a speedy response.

This class has great material, so HAVE FUN!