Course Information

CSE 350: Advanced Data Structures and Indexes - 4 Credits

Prerequisites

CSE 250 and 220. Computer Science, Computer Engineering, or Bioinformatics majors only

Course Description

Expands on CSE 250 by introducing techniques for data organization that account for the memory hierarchy and the need for concurrent access. Topics include relational data model and SQL, IO Complexity, On-Disk Tree- and Hash- based structures, Write-optimized data structures (e.g., LSM Indexes and Beta-Epsilon Trees), Serialization/Data Layout, Caching, Secondary Indexes, Concurrent Data Structures, and Versioned Data Structures. Introduces SQL queries and Relational Algebra as a framework for reasoning about data access methods.

Course Learning Outcomes

Upon completion of the course, students will be able to …

- compute the I/O complexity of an external-memory algorithm. (CS: 1, 2, 3, 6)
- identify the functionality of basic on-disk and concurrent data structures (CS: 1, 2)
- identify the trade-offs between different on-disk and concurrent data structures for related tasks. (CS: 1, 2)
- understand basic SQL, including filters, projection, joins, and aggregates, and how to design index structures to support each (CS: 1, 2, 3, 6)
- implement and understand basic external-memory algorithms (CS: 1, 2, 6)

(ABET Computer Science Student Outcomes listed with each learning outcome)

Textbook

There is no textbook for this course. All necessary materials will be made available through the course website.

Course Schedule (see course website for updated schedule, the following are typical topics that could be covered in the course)

Lectures	Topic	Assignments
1-3	Introduction, Math background, Introduction to relational model & SQL; Introduction to the C++ (for programming assignments), POSIX I/O interfaces	P0: Repository and dev environment setup & POSIX I/O
4	Recap on the RAM model; Review Array, LinkedList, Hash Table, Balanced Trees, runtime analysis on insertion/deletion/search in array/linked list/BST on RAM	P1: Binary search in RAM & EM
5	Intro to the External Memory (EM) model, runtime analysis of insertion/deletion/search in array/linked list/BST on EM	W1: Complexity analysis of RAM & EM data structures

Lectures	Topic	Assignments
6-8	Intro to B-tree on EM, insertion/deletion in B-tree	P2: Updates in B-tree
9-10	External sorting	P3: External sorting
11-12	External hashing technique (extendible/linear hashing)	W2: B-tree, hashing & external sorting
13-14	LSM Indexes, bloom filter	
15-16	Beta-Epsilon Trees	P4: Single-threaded LSM
17-19	Record, page and file layout in DBMS; Columnar Storage	
20-22	Buffer & space management	W3: Data layout & buffer management
23-24	Review mutex and other synchronization primitives; Concurrency on indexes; Hierarchical Locks	P5: Concurrency support for B-tree (OLC) or LSM (parallel merging)
25-26	Clustered vs Unclustered Indexes	
27-28	Versioned (immutable) Data Structures	W4: Data access & query cost analysis
29-30	Buffer Time / Advanced Topics / Review	

Computing Resources

You will be using various free on-line tools for this course – links will be provided on the course webpage. Course related communications should be through the Piazza forum linked through the course website. Piazza posts can be either public to the class, or private to instructors. Any email communications must come from your UB email account and include [CSE 350] in the subject line. All communications with course staff are expected to be professional. Graded work will be both submitted and returned via a dedicated student server minsky.cse.buffalo.edu.

Assignments

Assignments for this class will generally consist of :

- Programming Assignments
- In-class quizzes

Programming assignments will generally take 2-3 weeks to complete, including development of the project and any testing components. Written reflections will be assigned at the

completion of each programming assignment and will take approximately 1 week to complete. Expectations will be clearly noted when the assignments come out as well as the duration of the assignment. Please pay attention to the amount of time that each assignment provides and begin early. There will be roughly 4 programming assignments and written reflections each, in addition to an academic integrity assessment assignment and setup assignment. The exact number may vary slightly. **You must score a 100% on the Academic Integrity and Setup assignments or you will be given an F in the course**.

For in-class quizzes, we will only be accepting hand-written submissions during the lectures.

Late Policy for Programming Assignments

The policy for programming assignments is as follows. You have 5 grace days to use in total and you may only use 2 grace days for each project's final submission. You will always be graded on your latest submission. If your latest submission was submitted:

- On or before the deadline: you get 100% of what you earn.
- Late penalty per 24 hours for final submission of a project and you have not depleted all the 5 grace days: no penalty.
- >2 days late for final submission or > 0 day late for intermediate checkpoints: 0 points.

From this policy, you will see that all assignments must be submitted within two days past the assigned deadline. Each submission receives a grade based on the date it was submitted. Your final score is the score earned by your most recent submission.

Please plan accordingly. You will not be able to recover a grace day if you decide to work late and your score is not sufficiently higher. Grace days are automatically applied to the first instances of late submissions, and are non-refundable. For example, if an assignment is due on a Friday and you make a submission on Saturday, you will automatically use a grace day, regardless of whether you perform better or not. Be sure to test your code before submitting, especially with late submissions, in order to avoid wasting grace days.

Keep track of the time if you are working up until the deadline. Submissions become late after the set deadline, even by 1 minute. Keep in mind that submissions will close 48 hours after the original deadline and you will no longer be able to submit your code after that time.

Attendance and Participation

Attendance in lecture is not mandatory, but highly encouraged. There will be no lecture recording.

Recitations

We will be meeting for recitation to go over homework assignments and any questions you have about the material. In addition, the recitations may review or extend lectures and are an excellent environment to ask more individual questions regarding the course material. Attendance in recitation is mandatory. Questions posed in recitation may be graded for participation or correctness. You may miss 3 recitations without penalty for any reason.

Exams

There will be one in-class 120-minute midterm exam and one 120-minute final exam. The midterm is worth 15% of your grade each. The final exam is worth 15% of your grade. We reserve the right to change the scaling of the exams. No makeup exams will be given except in provably extreme circumstances.

Grading Policy

The following indicates the grade breakdown which will be used in assigning grades in the course. It is possible that these ranges may be adjusted at the end of the semester to address inconsistencies or hardships that arise. Grades will not be curved/adjusted during the semester.

Requirement	Weight
Assignments	50%
Quizzes	20%
Midterm Exam	15%
Final Exam	15%

Percentage	Letter Grade	Percentage	Letter Grade
[90,100]	A	[40,50)	C+
[80,90)	A-	[30,40)	C
[70,80)	B+	[20,30)	C-
[60,70)	B	[10,20)	D
[50,60)	B-	[0,10)	F

Regrading

Any questions about the grading of a piece of work must be raised **within one week** of the date that the graded work was returned to you.

Incomplete (I) grades

A grade of incomplete ("I") indicates that additional coursework is required to fulfill the requirements of a given course. Students may only be given an "I" grade if they have a passing average in coursework that has been completed and have well-defined parameters to complete the course requirements that could result in a grade better than the default grade. An "I" grade may not be assigned to a student who did not attend the course.

Prior to the end of the semester, students must initiate the request for an "I" grade and receive the instructor's approval. Assignment of an "I" grade is at the discretion of the instructor. The last day to resign the course is

Academic Integrity

Academic integrity is a fundamental university value. Through the honest completion of academic work, students sustain the integrity of the university while facilitating the

university's imperative for the transmission of knowledge and culture based upon the generation of new and innovative ideas. Please refer to the university Undergraduate Academic Integrity Policy (https://catalogs.buffalo.edu/content.php?catoid=1&navoid=19#academic-integrity) for additional information.

As an engineer or computer scientist, you have special ethical obligations. As per the NSPE Code of Ethics, “engineers shall avoid deceptive acts” and “shall conduct themselves honorably, responsibly, ethically, and lawfully so as to enhance the honor, reputation, and usefulness of the profession (https://www.nspe.org/resources/ethics/code-ethics). Similar sentiments of honesty, integrity, fairness, and responsibility are fundamental to the ACM Code of Ethics (https://www.acm.org/code-of-ethics).

A violation in this class generally results in an F for the entire course. The Computer Science and Engineering department's policy on academic integrity can be found here:

https://engineering.buffalo.edu/computer-science-engineering/information-for-students/undergraduate-program/cse-undergraduate-academic-policies/cse-academic-integrity-policy.html

What Constitutes a Violation of Academic Integrity?

These bullets should be obvious things not to do (but commonly occur):

- Turning in your friend’s code/write-up (obvious).
- Turning in solutions you found on Google with all the variable names changed (should be obvious). This is a copyright violation, in addition to an AI violation.
- Turning in solutions you found on Google with all the variable names changed and 2 lines added (should be obvious). This is also a copyright violation.
- Paying someone to do your work. You may as well not submit the work since you will fail the exams and the course.
- Posting to forums asking someone to solve the problem. Note: Aggregating every [stack overflow answer|result from google|other source] because you "understand it" will likely result in full credit on assignments (if you aren't caught) and then failure on every exam. Exams don't test if you know how to use Google, but rather test your understanding (i.e., can you understand the problems to arrive at a solution on your own). Also, other students are likely doing the same thing and then you will be wondering why 10 people that you don’t know have your solution.

Other violations that may not be as obvious:

- Working with a tutor who solves the assignment with you. If you have a tutor, please contact your instructor so that they may discuss with them what help is allowed.
- Sending your code to a friend to help them. If another student uses/submits your code, you are also liable and will be punished.
- Joining a chatroom for the course where someone posts their code once they finish, with the honor code that everyone needs to change it, in order to use it.
- Reading your friend’s code the night before it is due because you just need one more line to get everything working. It will most likely influence you directly or subconsciously to solve the problem identically, and your friend will also end up in trouble.
- Using generative AI software (e.g., ChatGPT, Cursor, Github Copilot, etc.) to obtain assistance or to directly obtain solutions for any part of the programming and written assignments. Answers produced by generative AI are based on its training data

collected from a variety of sources, so it is hard to determine whether it is an allowed resource and provide correct attribution/citation.

What Collaboration is Allowed?

Assignments in this course should be solved individually with only assistance from course staff and allowed resources. You may discuss and help one another with technical issues, such as how to get your compiler running, etc.

There is a gray area when it comes to discussing the problems with your peers and I do encourage you to work with one another to solve problems. That is the best way to learn and overcome obstacles. At the same time you need to be sure you do not overstep and not plagiarize. Talking out how you eventually reached the solution from a high level is okay:

"I used a stack to store the data and then looked for the value to return."

but explaining every step in detail/pseudocode is not okay:

"I copied the file tutorial into my code at the start of the function, then created a stack and pushed all of the data onto the stack, and finished by popping the elements until the value is found and use a return statement."

The first example is OK but the second is basically a summary of your code and is not acceptable, and remember that you shouldn't be showing any code at all for how to do any of it. Regardless of where you are working, you must always follow this rule: Never come away from discussions with your peers with any written work, either typed or photographed, and especially do not share or allow viewing of your written code.

What Resources are Allowed?

With all of this said, please feel free to use any [files|examples|tutorials] that we provide directly in your code (with proper attribution). Feel free to directly use anything from lectures or recitations. You will never be penalized for doing so, but should always provide attribution/citation for where you retrieved code from. Just remember, if you are citing an algorithm that is not provided by us, then you are probably overstepping.

More explicitly, you may use any of the following resources (with proper citation/attribution in your code):

- Any example files posted on the course webpage (from lecture or recitation).
- Any code that the instructor provides.
- Any code that the TAs provide.
- Any code from the cppreference (https://en.cppreference.com/w/), cplusplus (https://cplusplus.com/, excluding the forum), or learncpp (https://www.learncpp.com/, excluding the comments).

Omitting citation/attribution will result in an AI violation (and lawsuits later in life at your job). This is true even if you are using resources provided.

Amnesty Policy

We understand that students are under a lot of pressure and people make mistakes. If you have concerns that you may have violated academic integrity on a particular assignment, and would like to withdraw the assignment, you may do so by sending your instructor an email BEFORE THE VIOLATION IS DISCOVERED BY COURSE STAFF. The email should take the following format:

Dear Dr. Zhao,

I wish to inform you that on assignment X, the work I submitted was not entirely my own. I would like to withdraw my submission from consideration to preserve academic integrity.

J.Q. Student

Person #12345678

UBIT: jqstudent

When we receive this email, student J would receive a 0 on assignment X, but would not receive an F for the course, and would not be reported to the office of academic integrity.

Critical Campus Resources

Accessibility Resources

If you have any disability which requires reasonable accommodations to enable you to participate in this course, please contact the Office of Accessibility Resources in 60 Capen Hall, 716-645-2608 and also the instructor of this course during the first week of class. The office will provide you with information and review appropriate arrangements for reasonable accommodations, which can be found on the web at: http://www.buffalo.edu/studentlife/who-we-are/departments/accessibility.html.

Sexual Violence

UB is committed to providing a safe learning environment free of all forms of discrimination and sexual harassment, including sexual assault, domestic and dating violence and stalking. If you have experienced gender-based violence (intimate partner violence, attempted or completed sexual assault, harassment, coercion, stalking, etc.), UB has resources to help. This includes academic accommodations, health and counseling services, housing accommodations, helping with legal protective orders, and assistance with reporting the incident to police or other UB officials if you so choose. Please contact UB's Title IX Coordinator at 716-645-2266 for more information. For confidential assistance, you may also contact a Crisis Services Campus Advocate at 716-796-4399.

Mental Health

As a student you may experience a range of issues that can cause barriers to learning or reduce your ability to participate in daily activities. These might include strained relationships, anxiety, high levels of stress, alcohol/drug problems, feeling down, health concerns, or unwanted sexual experiences. Counseling, Health Services, and Health Promotion are here to help with these or other issues you may experience. You can learn more about these programs and services by contacting:

Counseling Services:

- 120 Richmond Quad (North Campus), 716-645-2720
- 202 Michael Hall (South Campus), 716-829-5800

Health Services:

- 4350 Maple Rd, Amherst, NY 14226, 716-829-3316

Health Promotion:

- 114 Student Union (North Campus), 716-645-2837

Diversity

The UB School of Engineering and Applied Sciences considers the diversity of its students, faculty, and staff to be a strength, critical to our success. We are committed to providing a safe space and a culture of mutual respect and inclusiveness for all. We believe a community of faculty, students, and staff who bring diverse life experiences and perspectives leads to a

superior working environment, and we welcome differences in race, ethnicity, gender, age, religion, language, intellectual and physical ability, sexual orientation, gender identity, socioeconomic status, and veteran status.