

# Midterm review

- Read instructions on blackboard (Page 1) of exam ahead of time.
- Scan the exam end to end first and plan your time before you begin answering the questions. . Short answer, true/false, algorithm interpretation/problem solving, complexity analysis, proofs, code interpretation questions.
- Id check:
  - Wait outside the exam hall at 2:15pm for id check (192 students enrolled).
  - 1 double-sided (8x11) cheat sheet allowed. Staff will check ids and cheat sheet.
- Closed book exam. No books or electronic devices of any kind allowed.
- Look at list of exam rules already posted on blackboard. This is Page 1 of your exam. You have to sign the AIB agreement.
- You should plan to stay in the exam hall until you hand in your answer sheet.

# Midterm review

- Material:
  - Lecture notes including all of Lecture 12 (including today's material) and associated text book readings.
  - **You have to read the textbook.**
  - **Code used in lectures is from the textbook – look it up and study it!**
  - Slides I emphasized during lecture.
  - Assignments.
  - Blackboard FAQs.
  - Try out some text book questions

## Note

- This is a 300-level course. Prepare as you would should for an advanced level course
  - Dig deep: Invest the appropriate amount of time digging into the details.
  - For each topic, know the why, what, complexity analysis, strengths, weaknesses.
  - Enjoy the process.
- Special emphasis to examples/topics/slides I asked you to pay special attention to during lectures.
- The following slides are just a **high-level guide** to help you organize information you should know. You are still expected to learn the details.

# C++

- Topics:
  - Initialization list, explicit constructor, Constant member function ((3 ways to use)
  - Pointers
  - L-Values, R-Values, References : Why, What, How?
  - 4 ways to pass parameters: call by -value/-constant-reference/reference/r-value reference examples and usage (Lecture 1, 2 and 4)
  - Return passing: return by -value/-constant-reference/reference examples and usage
  - Copy vs move, Std::swap, std::move: deep copy vs shallow copy
  - Big Five
- IntCell() class, FindMax() example all aspects discussed
- Function template, Comparable, Function objects
- Matrices: all aspects discussed, operator[]
- Make a list of STL concepts you have learned for future reference.

# Complexity

- Relative rate of function growth.
- Useful rules of Big-O.
- Determine complexity for problems similar in scope to examples discussed in class:
  - Factorial, Fibonacci, MaxSubSequence (various ways and runtime analysis of each), binary search etc.
- Limitations
- Try out some textbook questions

# Lists, Stacks, Queues

- What? Why? How to implement? For each topic below
  - Vectors vs Lists
  - Begin(), end(), erase
  - Iterators, const\_iterators

# Trees

- In general, What are they? Why are they needed? How is it implemented?
- Where does it fail (this should motivate the next topic discussed in the Trees -related lecture?
  - Build on the summary slides and make sure you connect the thread between topics for yourself as I did in class.
- Expression trees, Binary Trees, Binary Search trees, AVL trees, Splay trees, B-trees, M-ary tree, Intro to Sets and Maps etc.
  - Average case analysis of BST
  - Contains, FindMin, FindMax, insert, remove, Pair function
  - Amortized cost computation
  - Complexity/Running time schemes for various operations (insert, delete, find, , ordering operations where appropriate) discussed in lectures

# Hashing

- Hash functions
- Separate Chaining (details and analysis)
- Linear probing, Quadratic Probing, Double Hashing
- Hashing Topics not covered in class on 10-17-22 will not be on the test



# Examples of Analysis and Proofs

The following are only examples. Any analysis or proofs discussed in any of the 12 lectures is fair game for the exam.

Examples:

- Lecture 4 and 5: Analysis
- Lecture 12: Theorems and Proofs.