CS367: *Introduction to Data Structures*

EXAM INFORMATION - Lecture 1, Spring 2018

In this page: Midterm Exam | Final Exam
Related pages: Home | Exams | Exam Format

Midterm Exam

- **Date:** Thursday, March 15, 2018
- **Location:** To be Determined
- **Time:** 7:15 - 9:15 pm
- **What to bring:** You may bring one 8.5" x 11" page of notes with information on one side of the page only. The page of notes may be either printed or hand written. Your name should be printed somewhere on the page and you must turn in your page of notes with your exam.

Students are responsible for material covered in:

- Lectures (through material covered on Tuesday, October 24)
- Readings: Introduction, Lists, Exceptions, Linked Lists, Complexity, Stacks and Queues, Priority Queues, Intro to Trees
- Homeworks: 1 - 3
- Programming assignments: 1 - 2

Note that this reference page will be provided in the exam.

A selection of sample exam questions (and solutions) will be made available prior to the exam..

- **Midterm exam topics**

Topics List:

- **General**
  Abstract data type (ADT), data structure, Java interfaces (defining, implementing, using an interface as a type), using Java generics, primitive vs. reference types in Java

- **Lists**
  List ADT, List operations, implementing List using an array, implementing List using a chain of linked nodes, complexities of List operations for different implementations, using a List

- **Iterators**
  What they are, how to use them, how to implement one for an ADT when the underlying data structure is an array or a chain of linked nodes

- **Java Exceptions**
  Defining new exceptions, checked vs. unchecked, throwing an exceptions using `throw`, handling an exception using `try-catch`, handling an exception by passing it down the call stack, listing exceptions in a `throws` clause

- **Chains of Linked Nodes**
  List node, operations on chains of nodes, head pointer, tail pointer, dummy header node, singly linked chains, doubly linked chains, circular linked chains, comparison of chain variations

- **Complexity**
  What complexity is, problem size, constant time (O(1)), linear time (O(N)), quadratic time

(O(N^2)), big-O notation, worst-case vs. best/average cases, determining complexity of Java code, limitations of complexity analysis

- **Stacks**
  Stack ADT, Stack operations, implementing Stack using an array, implementing Stack using chain of nodes variations, complexities of Stack operations for different implementations, using a Stack

- **Queues**
  Queue ADT, Queue operations, implementing Queue using array variations, circular arrays, implementing Queue using chain of nodes variations, complexities of Queue operations for different implementations, using a Queue

- **Recursion**
  Recursive methods, base case, recursion rules, how method calls work (i.e., activation records and the call stack), recursion vs. iteration, recursive data structures, analyzing complexity of recursive methods, recurrence equations, determining complexity using recurrence relations

- **Intro to Trees**
  Position-oriented vs. value-oriented ADTs, tree terminology (root, leaf, node, edge, parent, child, sibling, ancestor, descendant, subtree, path, length of a path, height of a tree, full tree, complete tree, height-balanced tree, general vs binary trees)

Final Exam

- **Date:** Wednesday, May 9, 2018
- **Time:** 12:25 - 9:25 pm
- **Location:**132 Noland
- **What to bring:** You may bring one 8.5" x 11" page of notes with information on both sides of the page. The page of notes may be either printed or hand written. Your name should be printed somewhere on the page and you must turn in your page of notes with your exam.

Students are responsible for material covered in:

- Lectures
- Readings
- Homeworks
- Programming assignments

Note that this reference page will be provided in the exam.

A selection of sample exam questions (and solutions) will be made available prior to the exam.

Topics List:

- **Final exam topics**
  The final is cumulative although there will be much greater emphasis on the topics covered since the midterm.
- **Search**
  Java interfaces (defining, implementing, using an interface as a type), the Comparable interface, sequential search, binary search, complexities of search algorithms

- **Trees**
  General trees, binary trees, implementation of trees, writing recursive methods that traverse a tree, pre-order traversal, post-order traversal, level-order traversal, in-order traversal

- **Binary Search Trees**
  Binary search trees (BSTs), key value, implementing BSTs, algorithms for BST operations (print, lookup, insert, delete), in-order successor, in-order predecessor, complexities of BST operations

- **Red-Black Trees**
  Balanced search trees, red-black trees, red-black tree properties, root property, red property, black property, algorithms for red-black tree operations (print, lookup, insert)

- **Priority queues**
  Priority Queue ADT, implementing Priority Queue (using a heap, unsorted array, sorted chain of nodes, etc.) and implications for complexities, heap operations (algorithms & complexities), array implementation of heaps

- **Sorting**
  Comparison sort, bubble sort, insertion sort, selection sort, merge sort, quick sort, heap sort, stable sort, radix sort, complexities of sorts (both in terms of time and space), what input results in the best or worst case complexity

- **Graphs**
  Graph terminology (node, edge, directed, undirected, source, target, adjacent, successor, degree, in-degree, out-degree, path, acyclic, cyclic, complete graph, connected graph, weakly-connected graph, strongly-connected graph, weighted graph, DAG), graph ADT (implementations & complexities), representing edges using adjacency lists, representing edges using an adjacency matrix, graph traversals (BFS & DFS), applications of graph traversals (finding paths, topological ordering)

- **Hashing**
  Hashing terminology (hash table, hash function, key, load factor, collision, ideal hashing), hash table operations (algorithms & complexities),

- **Data Structures in Computer Sustems**
  Binary numbers, computer organization, instruction format and execution, memory systems, interrupts, operating systems, virtual memory

Last Updated: 1/18/2018 ©2018 Beck Hasti and Charles Fischer