Final Review
When and Where

• When
  • December 8th from 7:30pm to 10:00pm

• Where
  • STC 1012
General Information

• Question types are similar to midterm exam types
  • multiple choices 20%
  • short answers 20%
  • Others on design, analysis, implementation and application of data structures and algorithms
• ~ 1/3 on topics before midterm
• ~ 2/3 on topics after midterm
• Office hours:
  • Dec 5 and Dec 7: 12pm – 2pm, DC 2129
  • Dec 6: 1:30 – 2:30 pm, Dec 8: 2pm – 3pm, MC 4065
Overview

• List of Data Structures (ADTs)
  • Linear Data Structures
  • Non-linear Data Structures

• Operations on data structures

• Sorting Algorithms

• Recursion
Linear Data Structures

• Array
  • unsorted array
  • sorted array

• Python List
  • unsorted Python List
  • sorted Python List

• Linked Lists (sorted and unsorted)
  • Singly linked list
  • Doubly linked list
  • Circular linked list

• Stack

• Queue
  • Priority Queue
Non-Linear Data Structures

• Hash Table
  • Open Addressing
    • Linear Probing
    • Quadratic Probing
    • Double Hashing
  • Separate Chaining

• Tree
  • Binary Tree
    • Full, Perfect, Complete (Heap)
    • Binary Search Tree
      • Self-Balancing Binary Search Tree: AVL Tree

• Graph
  • Adjacency Matrix
  • Adjacency Lists
Common Operations on Data Structures

- Insertion
- Deletion
- Search
  - Any
  - All (traversal)
  - Minimum
  - Maximum
  - Predecessor
  - Successor
  - ……

Consider building up a table for
(Data Structures x Operations)

Ask yourself:
1. Is the data structure implemented by other data structure(s)?
2. do you know how to achieve Operation x on Data Structure y implemented by Data Structure z?
3. 2. do you know the efficiency of Operation x on Data Structure y implemented by Data Structure z?
4. …
Sorting Algorithms

• Bubble Sort
• Selection Sort
• Insertion Sort
• Merge Sort
• Quick Sort
• Radix Sort
• Heap Sort
• BST Sort

Ask yourself:
1. How does it work?
2. What data structure(s) fit this sorting algorithm? Is there any difference?
3. Is it a in-place sorting algorithm?
4. What is its time complexity?
5. How much extra space does it need?
6. ...
Recursion

• How to use recursion to solve problems?
  • divide problem into smaller problems
  • find base case
  • find recursive case
  • make progress towards the base case

• Use recursive call trees to help with tracing function calls

• Single recursion vs. Multiple Recursion

• Recurrence Equation

• Applications:
  • Binary Search
  • Towers of Hanoi