CS151 Intro to Data Structures

Midterm Review

Announcements

HW04 Due Sunday

- Binary trees will be on your exam!

Exam Wednesday

Exam Format

- 1 page 8.5/11in cheat sheet allowed (front and back)
- Format:
  - 11 points T/F questions
  - 8 points reading and understanding code
  - 51 points open ended programming
    - I won’t be harsh on syntax grading
      - I won’t try to compile your code :)
    - Be careful with types! You’ll get points deducted if you code doesn’t type check
- Tips:
  - For the coding portion, DRAW! It will help you

Topics

Data Structures

- Arrays
- Expandable Arrays
- Stacks
- Queues
- Linked Lists
- Binary Trees

Other concepts:

- Generics
- Iterators
- Big-O analysis
- OOP & Inheritance
- Interfaces

Data Structures

Expandable Arrays

search

- How do we implement?
- Best case?
- Worst case?

insertion

- How do we implement?
- Best case?
- Worst case?

removal

- How do we implement?
- Best Case?
- Worst Case?

LinkedList

search

- How do we implement?
- Best case?
- Worst case?

insertion

- How do we implement?
- Best case?
- Worst case?

removal

- How do we implement?
- Best Case?
- Worst Case?

Stacks - LinkedList implementation

Search?

- How do we implement?

Insertion?

- How do we implement?
- Best Case?
- Worst Case?

Removal?

- How do we implement?
- Best Case?
- Worst Case?

Stacks - Array implementation

Search?

- How do we implement?

Insertion?

- How do we implement?
- Best Case?
- Worst Case?

Removal?

- How do we implement?
- Best Case?
- Worst Case?

Queues - LinkedList implementation

Search?

- How do we implement?

Insertion?

- How do we implement?
- Best Case?
- Worst Case?

Removal?

- How do we implement?
- Best Case?
- Worst Case?

Queues - Array implementation

Search?

- How do we implement?

Insertion?

- How do we implement?
- Best Case?
- Worst Case?

Removal?

- How do we implement?
- Best Case?
- Worst Case?

Binary Trees

Search?

- How do we implement?
- Best Case?
- Worst Case?

Insertion?

- How do we implement?
- Best Case?
- Worst Case?

Removal?

- How do we implement?
- Best Case?
- Worst Case?

Binary ***SEARCH*** Trees

(assume balanced)

Search?

- How do we implement?
- Best Case?
- Worst Case?

Insertion?

- How do we implement?
- Best Case?
- Worst Case?

Removal?

- How do we implement?
- Best Case?
- Worst Case?

Other Concepts

Generics

What is a generic?

How do we declare a generic class?

What can a generic class hold?

Iterators

What methods can we call on iterators?

Advantages / disadvantages of iterators vs loops?

Runtime Complexity

Sort these from fastest to slowest:

- O(n)
- O(n^2)
- O(logn)
- O(1)
- O(2^n)

Coding Questions

Coding Question #1

You are given a string containing a combination of square brackets [ ], curly braces { }, and parentheses ( ). Use a stack to determine if the input string is valid in terms of bracket balancing. **Use a stack**

"[{ }]" => True
"[{] }" => False
"{ [( )] }" => True
"{ [( ]) }" => False

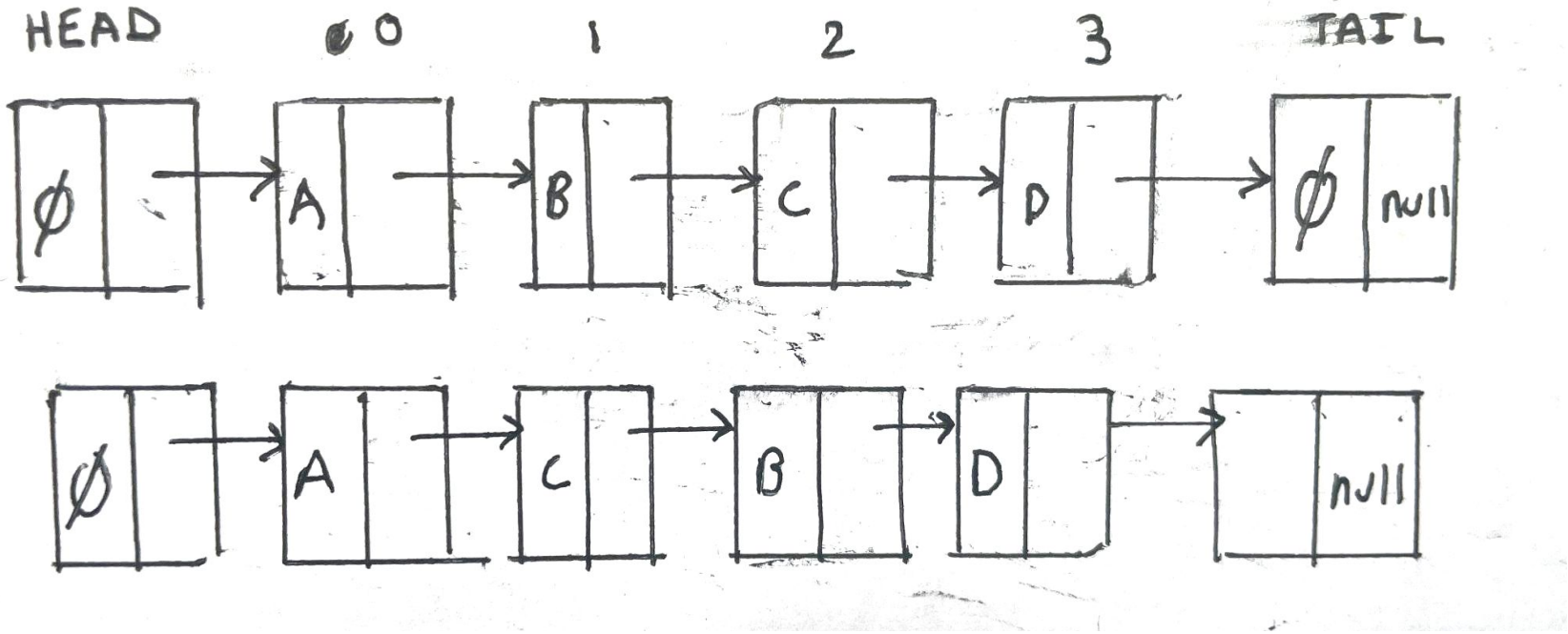
Coding Question #1

Runtime complexity?

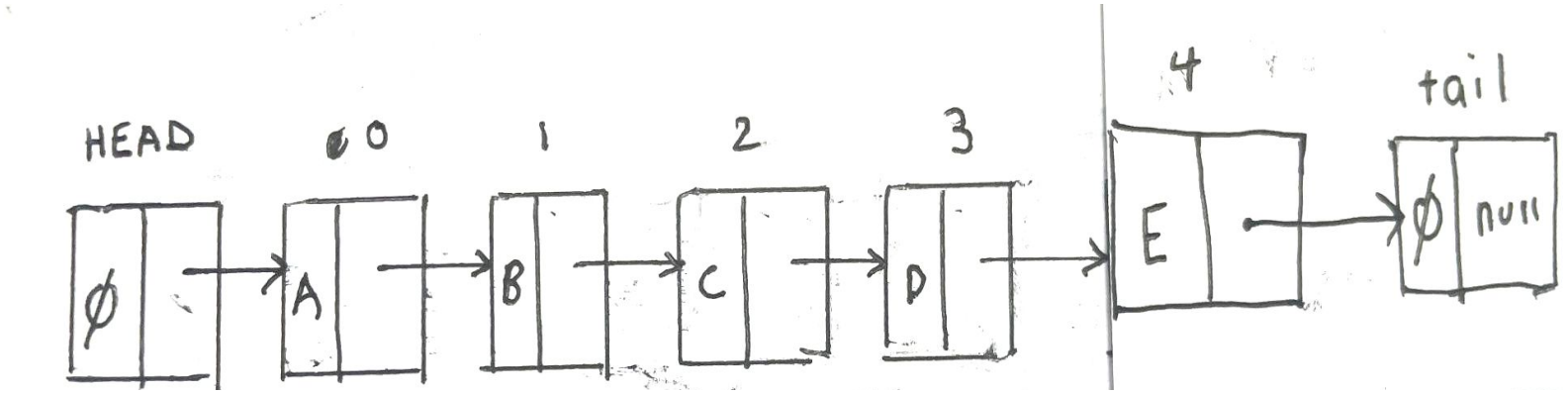
Memory complexity?

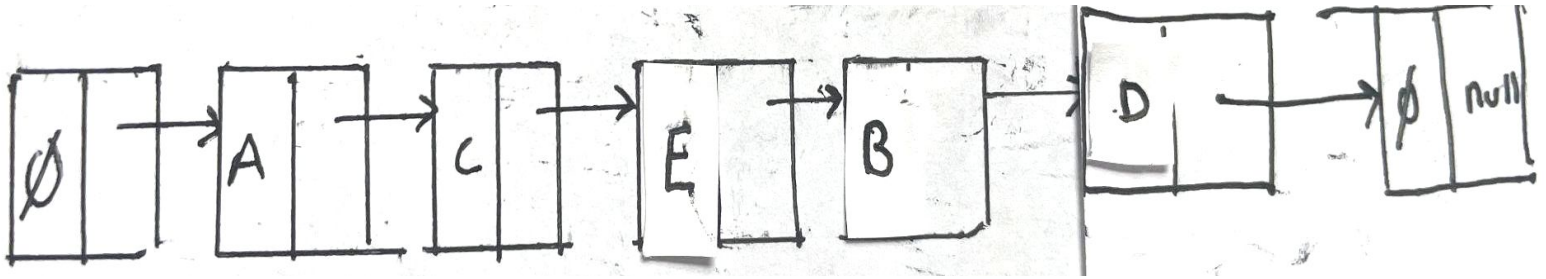
Coding Question #2

`rearrangeEvenOdd` modify the linked list in such a way that nodes with even indices (0, 2, 4, ...) appear before nodes with odd indices (1, 3, 5, ...). Ensure that the rearrangement is done in-place.



Coding Question #2





Coding Question #2

Edge cases!

Head -> Tail

Head -> A -> Tail

Head -> A -> B -> Tail

Do we handle these?

Coding Question #2

Runtime complexity?

Memory complexity?

Coding Question 3:

name,	**intensity, kind**
splash,	50, water
fireball,	100, fire
ignite,	15, fire
terraform,	20, earth

Design four classes: `water`, `fire`, `earth`, and `spell`.
Write a method **`castSpells`** takes two `Spells` and returns a `int` indicating which spell won

Water always beats fire regardless of intensity

Otherwise, compare intensity