WOMEN WHO CODE TORONTO

Implementing Linked List In Ruby

Rebecca Qu

Wifi: Mozilla Guest Password: (no password needed)

Thank you

@WomenWhoCodeTO

Next Meetup

intelliware.ca

software development

@WomenWhoCodeTO

#RandomActsOfPizza

HAVE YOU JOINED SLACK?

Get updates first! Special giveaways on Slack

bit.ly/WWCTOSlack

@WomenWhoCodeTO

COMMUNITY UPDATES

SEPT. 25-26

FITC Web Unleashed

Code: **WWC** ($150 off)

www.webunleashed.ca

NOV. 13-15

SecTor

Code: **WWCT2017** (10% off)

www.sector.ca

@WomenWhoCodeTO

A Few Words

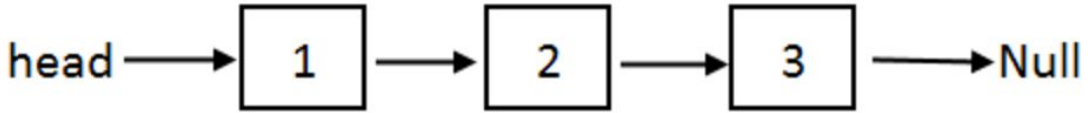
Implementing Linked Lists in Ruby

WWC TO: Algorithms Study Night

 @BeksQu

Linked Lists

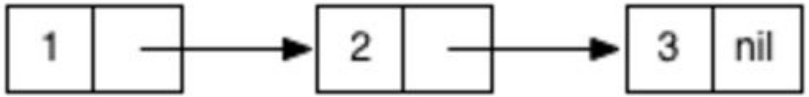
- data structure
- a sequence of nodes
- Singly-Linked List: each node points to the next node in the list



- Doubly-Linked List: each node points to next & previous node in the list

Nodes

- Each node contains: value & pointer



Node values within a list:

- can contain anything - strings, characters, numbers, etc.
- can be sorted or unsorted
- can be unique or duplicates

Why do we care about Linked Lists?

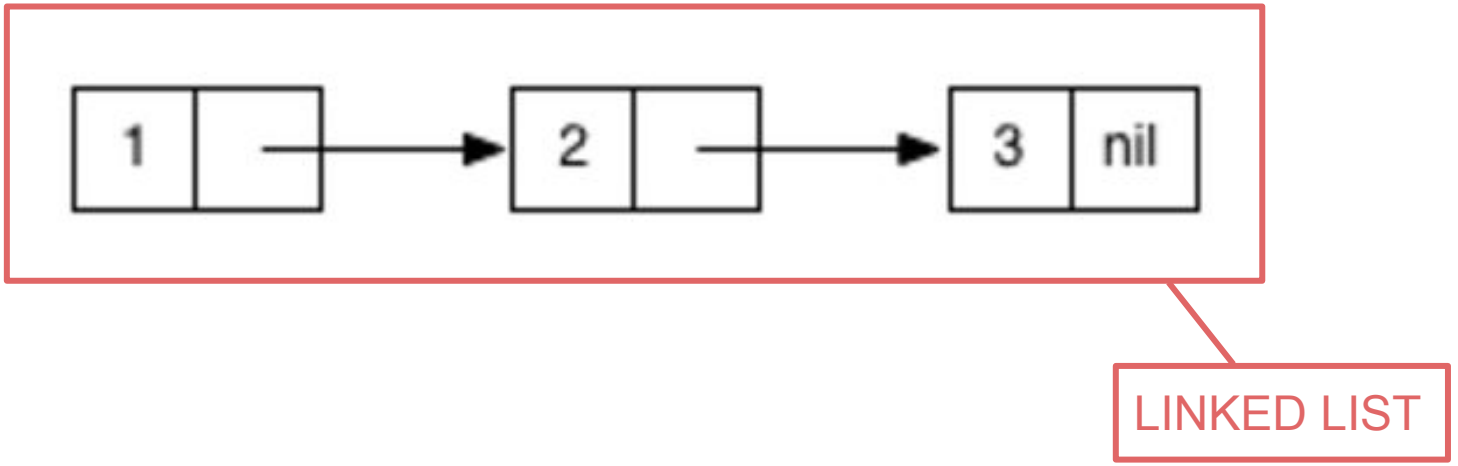
Advantages:

- Dynamic data structure: no memory wasted
- Insertion and deletion to beginning of list is constant time O(1)

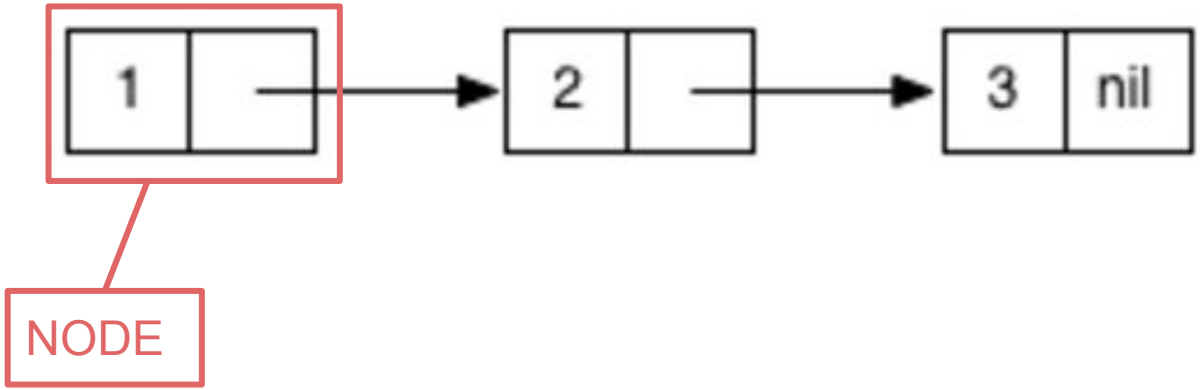
Disadvantages:

- Memory usage: node pointer requires extra memory
- Accessing elements (other than 1st) takes linear time O(n)

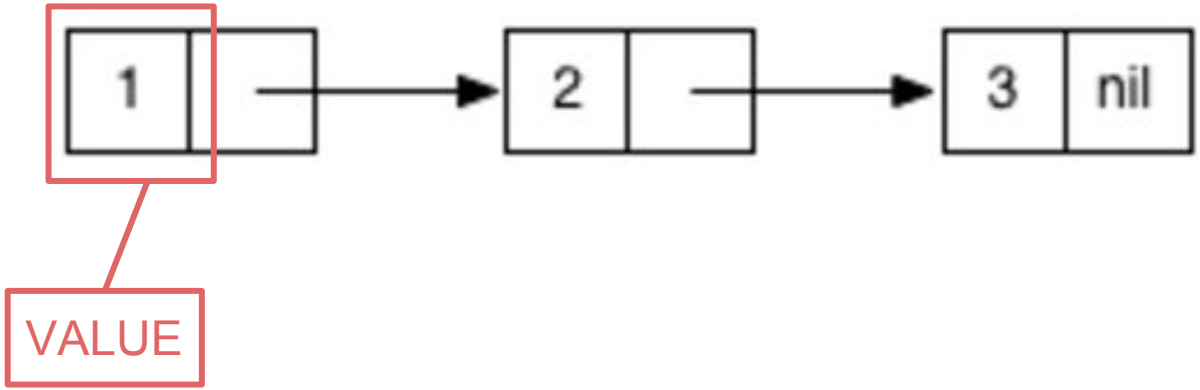
Anatomy a Linked List...



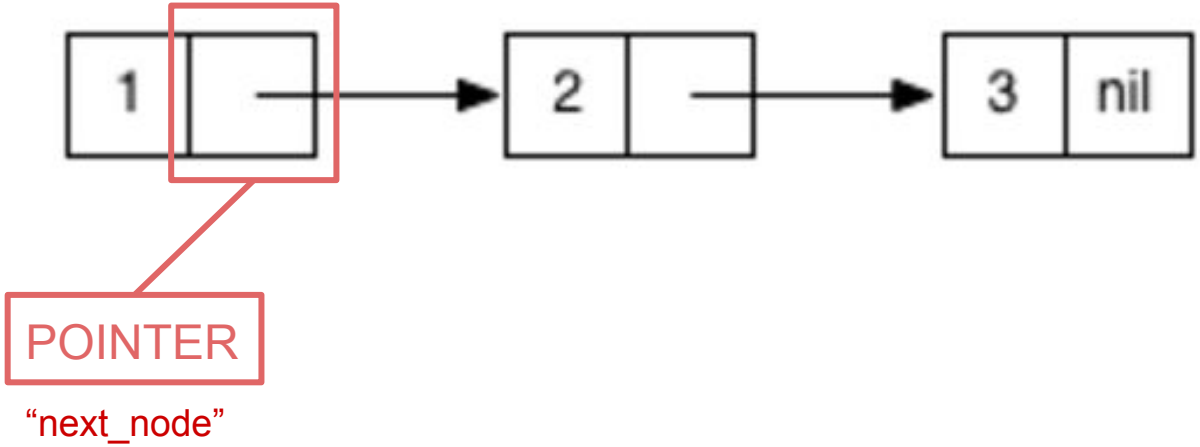
Anatomy a Linked List...



Anatomy a Linked List...



Anatomy a Linked List...



Anatomy a Linked List...

Now, let’s implement one!

Your turn...

https://github.com/RebeccaQu/AlgorithmsStudyNight

- **PREPEND**: add new node to start of list

APPEND: add new node to end of list

POP: removes last node in list

HEAD: returns first node in list

TAIL: returns last node in list

SIZE: returns total number of nodes in list

AT: returns node at a given index

CONTAINS: returns true if list contains a given value

FIND: returns index of node containing given data, or nil if not found

TO_S: represent your LinkedList objects as strings, so you can preview them in the console.

BONUS!

INSERT_AT: inserts node at given index

REMOVE_AT: removes node at given index

REVERSE: reverse the linked list

Additional Resources

1. Linked Lists in Plain English:

   https://www.youtube.com/watch?v=oiW79L8VYXk

2. Linked List: Ruby's Missing Data Structure:

   https://www.sitepoint.com/rubys-missing-data-structure/

3. A More Verbose Explanation with Plenty of Diagrams:

   http://www.cs.cmu.edu/~adamchik/15-121/lectures/Linked%20Lists/linked%20lists.html

Thank you!

Huge thanks to

Mozilla - you’re the best!!

Thank you Beks!

@WomenWhoCodeTO

WOMEN WHO CODE® TORONTO

@WomenWhoCodeTO

www.womenwhocode.com/donate

Wifi: Mozilla Guest Password: (no password needed)