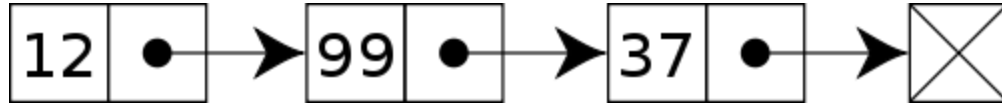


Study Write Up: Linked Lists

Linked lists are quite important and well-known in computer science. They are a type of data structure that looks like this:



Essentially, it is a list of data containing nodes. Each node has two parts to it: the data and a pointer to the location of the next node. Eventually, the list has a terminator node which, of course, terminates the list.

Linked lists have many potential applications. A simple example of a linked list is this: you have some errands to do, so you grab a piece of paper and write:

- bank
- groceries
- drycleaning

Then, you remember that you also need to buy stamps. You need to do that after the bank, though. You could copy your whole list onto a new piece of paper:

- bank
- stamps
- groceries
- drop off drycleaning

or you could write on the one you had:

- bank STAMPS
- groceries
- drop off drycleaning

As you think of other errands, you might write them at the bottom of the list, but with arrows reminding you what order to do them in. This is a linked list. It's quicker and easier than copying the whole list around every time you add something.

Then your cell phone rings while you're at the bank "Hey, I got the stamps, don't pick up any more". You just cross "STAMPS" off the list, you don't rewrite a whole new one without "STAMPS" in it. The idea here is that you want to add and remove lots of items, and order matters, but you don't want to recopy the whole list after each insertion or deletion.

Here is a simple linked list in Ruby:

```
class Node

  attr_accessor :data, :next

  def initialize(data)

    @data = data

  end

end

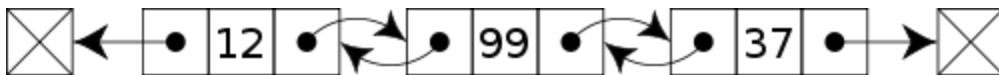
node_1 = Node.new("How are you doing?")
node_2 = Node.new("I'm fine, thanks.")

node_1.next = node_2

puts node_1.data

puts node_1.next.data
```

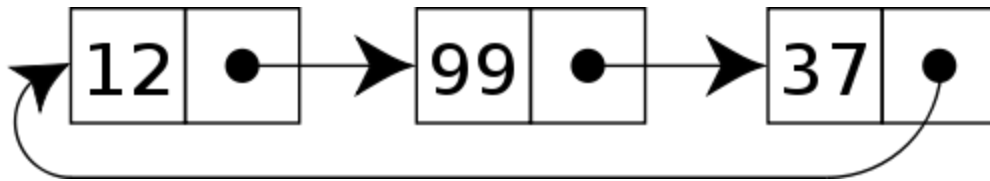
Additionally, there are different types of linked lists. The one we've been talking about so far is a singly linked list, as seen in the image at the top. However, there are also doubly linked lists, which look like so:



The idea behind a doubly linked list is to have two links in each node, one to the next node and one to the previous node.

In an “open” or “linear” linked list, the last node of the list contains a null or nil value, a special value used to indicate the lack of further nodes. A less common convention is to make it point

back to the first node of the list; in that case the list is said to be “circular”, which looks like this:



Conclusively, linked lists can be handy in various computer science situations and have been used many times in all sorts of scenarios.