







Search within a Linked List




Let's add a few more useful methods to our linked list class. Wouldn't it be useful if we could tell if our list was empty or not, as with our `Stack` and `Queue` classes?

We should also be able to find specific elements in our linked list. Traversing through data structures is something you'll want to get a lot of practice with! Let's create an `indexOf` method that takes an `element` as an argument, and returns that element's `index` in the linked list. If the element is not found in the linked list, return `-1`.

Let's also implement a method that does the opposite: an `elementAt` method that takes an `index` as an argument and returns the `element` at the given `index`. If no `element` is found, return `undefined`.

Write an `isEmpty` method that checks if the linked list is empty, an `indexOf` method that returns the `index` of a given element, and an `elementAt` that returns an `element` at a given `index`.

	Your <code>LinkedList</code> class should have an <code>isEmpty</code> method.
	Your <code>isEmpty</code> method should return <code>false</code> when there is at least one element in the linked list.
	Your <code>isEmpty</code> method should return <code>true</code> when there are no elements in the linked list.
	Your <code>LinkedList</code> class should have an <code>indexOf</code> method.
	Your <code>indexOf</code> method should return the index of a given element found in the linked list.
	Your <code>indexOf</code> method should return <code>-1</code> if the given element is not found in the linked list.

	Your <code>LinkedList</code> class should have an <code>elementAt</code> method.
	Your <code>elementAt</code> method should return the element found at a given index in the linked list.
	Your <code>elementAt</code> method should return <code>undefined</code> if the given element is not found at a given index in the linked list.