

---

# Linked Lists

Ms. Laura Burdick - Ph.D. Candidate, University of Michigan (and GCC alum '15!)

As you come in:

- If you want to follow along and take notes, you can find today's slides at <http://laura-burdick.github.io/linkedLists.pdf>.
- Jot down some ideas for the following questions:
  - What data structures have you learned about?
  - Why do we have so many different data structures?

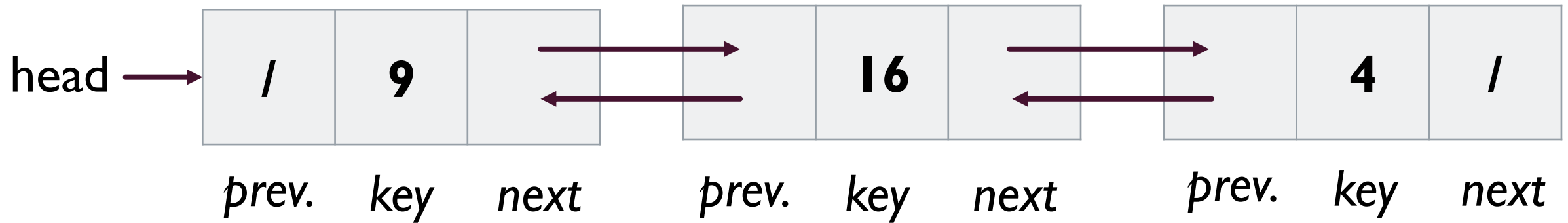
Brainstorm: What data structures have you learned about?

# Why do we have so many different data structures?

By the end of this lesson, you will be able to:

1. Implement a linked list in Java.
2. Explain how to implement the search and insert operations on linked lists.
3. Compare the efficiency of linked lists versus arrays on the search and insert operations.

# Linked Lists



- A linked list is made up of elements
- Each element contains three things:
  1. A key (what you want to store in the element)
  2. A reference to the previous element
  3. A reference to the next element

# References

- A reference holds the **memory address** of an item



Computer Memory

THE CLASSIC WORK  
NEWLY UPDATED AND REVISED

The Art of  
Computer  
Programming

VOLUME 3  
Sorting and Searching  
Second Edition

DONALD E. KNUTH

[The art of Computer Programming. Volume 2,  
Seminumerical Algorithms](#)

Knuth, Donald Ervin, 1938-  
Book - 1997 - 005.1 Kn  
1 On Shelf At: Downtown Library  
No requests on this item

Reference

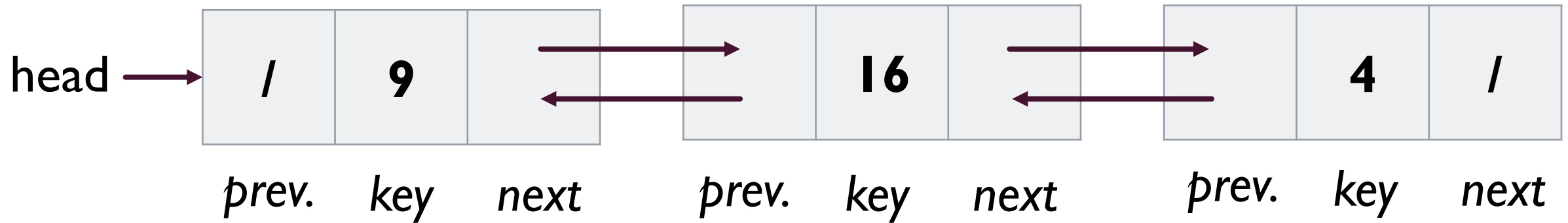
## References in Java

- Let's suppose we created a **LinkedList** class.

```
LinkedList myList = new LinkedList();
```

- **myList** holds a reference to a **LinkedList** object.

## References



- *Previous* holds the **memory address** of the previous element
- *Next* holds the **memory address** of the next element



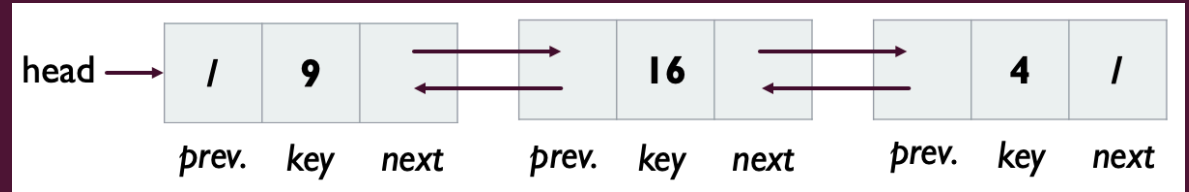
 Respond at **PollEv.com/lauraburdick889**

 Text **LAURABURDICK889** to **37607** once to join, then **A or B**

Do elements of a linked list need to be stored next to each other (contiguously) in memory?

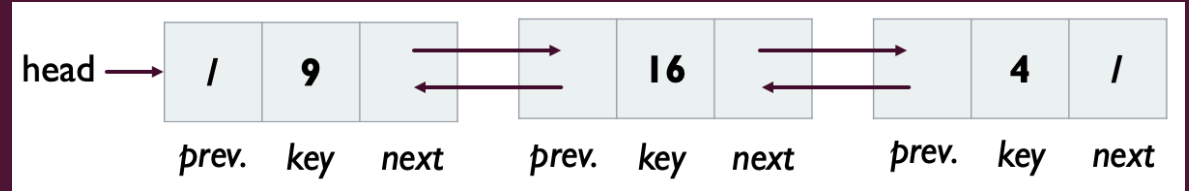
- A. Yes, elements need to be stored contiguously.
- B. No, elements do not need to be stored contiguously.

# Searching



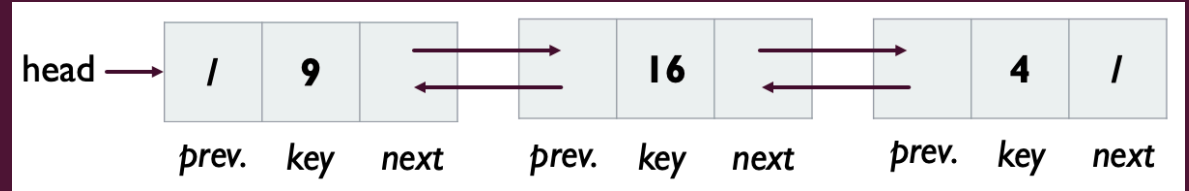
- **Input**: List  $L$ ; key  $k$  you are searching for (e.g., 4)
  - **Output**: A reference to the element that contains that key
1. Think about it silently and take notes (1 minute).
  2. Talk with someone next to you about your ideas (2 minutes).
  3. Be prepared to share your thoughts with the class!

## Searching



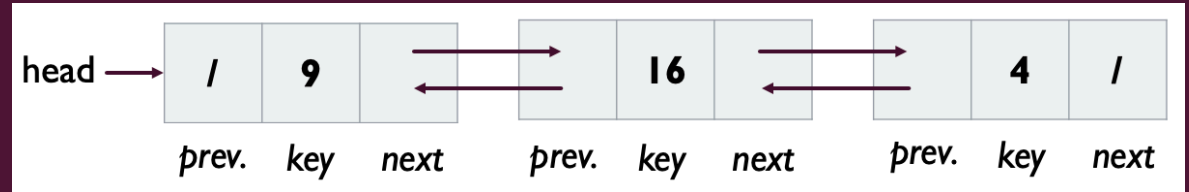
- **Input**: List  $L$ ; key  $k$  you are searching for (e.g., 4)
- **Output**: A reference to the element that contains that key

# Searching



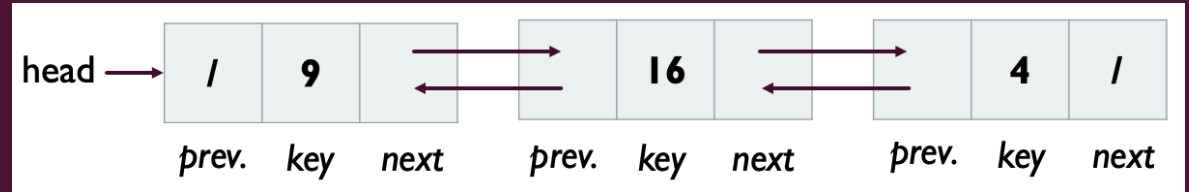
- **Input**: List  $L$ ; key  $k$  you are searching for (e.g., 4)
- **Output**: A reference to the element that contains that key

# Inserting



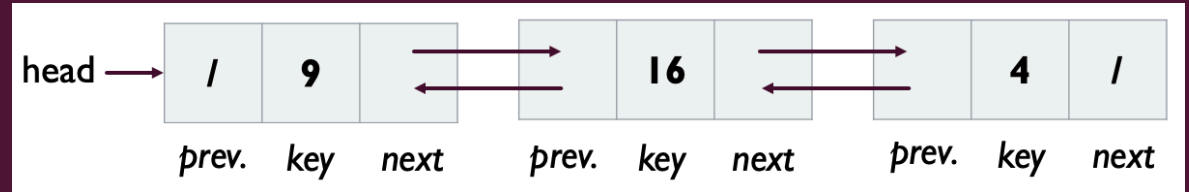
- **Input**: List  $L$ ; key  $k$  you want to insert (e.g., 4)
  - **Output**: Void
  - *(We'll insert the new element at the start of the linked list.)*
1. Think about it silently and take notes (1 minute).
  2. Talk with someone next to you about your ideas (2 minutes).
  3. Be prepared to share your thoughts with the class!

# Inserting



- **Input**: List  $L$ ; key  $k$  you want to insert (e.g., 4)
- **Output**: Void
- *(We'll insert the new element at the start of the linked list.)*

# Inserting



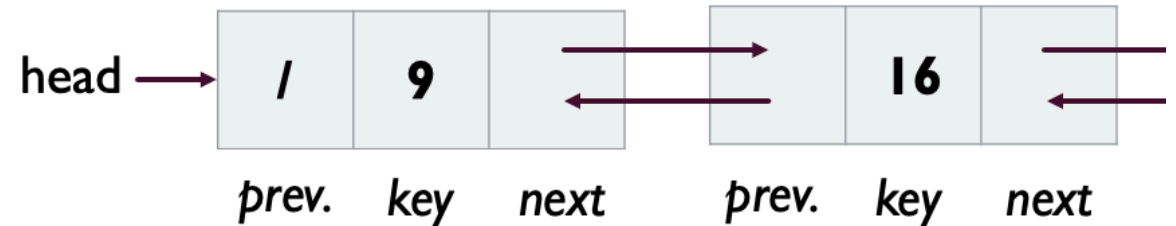
- **Input**: List  $L$ ; key  $k$  you want to insert (e.g., 4)
- **Output**: Void
- *(We'll insert the new element at the start of the linked list.)*

## Arrays

0	1	2
9	16	4

- Linear data structure
- Stored in contiguous memory
- Access items using indices
- Operations: search, insert, delete...

## Linked Lists



- Linear data structure
- Elements can be stored anywhere in memory
- Access items using references
- Operations: search, insert, delete...



 Respond at **PollEv.com/lauraburdick889**

 Text **LAURABURDICK889** to **37607** once to join, then **A or B**

On average, is it more efficient to **access** an item in an array or a linked list?

- A. Accessing an array element is more efficient.
- B. Accessing a linked list element is more efficient.
- C. Both are the same efficiency.
- D. We can't determine the efficiency.

 Respond at **PollEv.com/lauraburdick889**

 Text **LAURABURDICK889** to **37607** once to join, then **A or B**

On average, is it more efficient to **insert** an item in an array or a linked list?

- A. Inserting an array element is more efficient.
- B. Inserting a linked list element is more efficient.
- C. Both are the same efficiency.
- D. It depends.

## 1-minute Pause

- Stop and review the material we've covered so far (slides, notes, code, etc.).
- Is there anything that you need clarified?



# Let's do some programming!

## Let's do some programming!

With a partner, work on implementing the following functions inside the LinkedList class:

```
/* Add an element to the linked list
 * Input: The integer you want to add
 * Output: None
 */
public void addElement(int key) {}
```

```
/* Override the toString() method so that when you
 * print the list, all the values are printed out.
 * Input: None
 * Output: String to be printed
 */
public String toString() {}
```

If you have extra time, work on the following...

- Think about how you would delete an element from a linked list. Write out pseudocode first, and then implement the function (inside `LinkedList`). It should take as input the integer to delete, delete the first instance of that integer, and return void.
- Create a new class, `SinglyLinkedList`, which is the same as `LinkedList` except that it only has a next reference (and no previous reference). This is called a singly-linked list, in contrast to the doubly-linked list that we have looked at previously. Re-implement the `addElement`, `deleteElement`, and `toString` functions.

## Main Takeaways

1. A linked list is a data structure made up of elements. Each element contains a key, previous reference, and next reference.
2. Linked lists don't need to be stored contiguously in memory.
3. Linked lists are fast at inserting, but slow at searching (compared to arrays).
4. Having a large toolbox of data structures allows you to write efficient code in many different situations.

# More Questions?

E-mail me at [wenlaura@umich.edu](mailto:wenlaura@umich.edu).

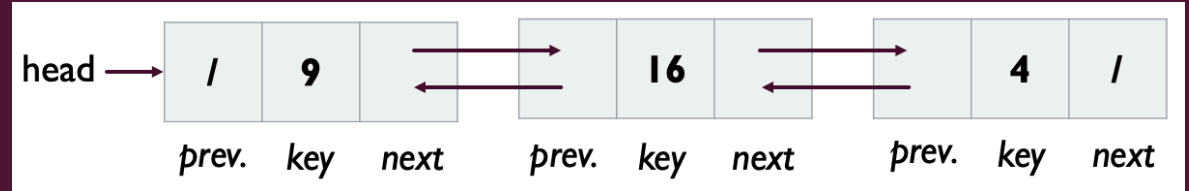


# Extra Slides!

For your additional enjoyment, or if we have extra time at the end of the lesson.

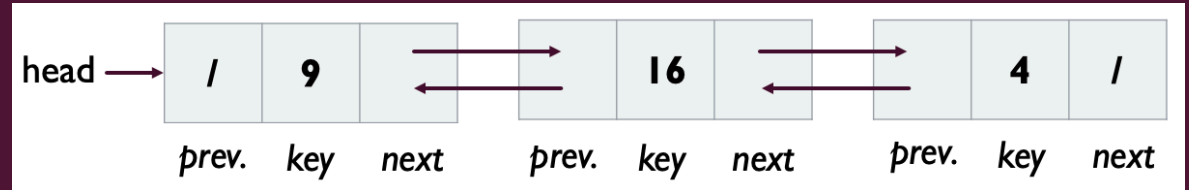


# Deleting



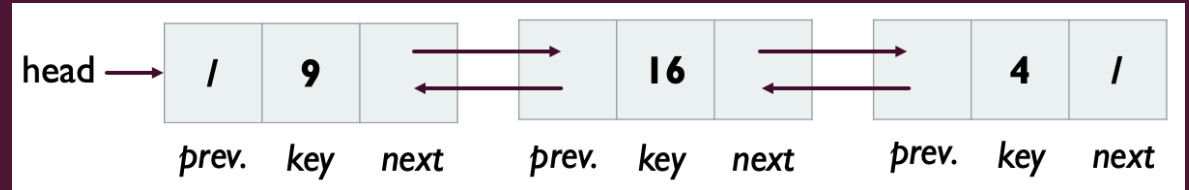
- **Input**: List  $L$ ; key  $k$  you want to delete (e.g., 4)
  - **Output**: void
1. Think about it silently and take notes (1 minute).
  2. Talk with someone next to you about your ideas (2 minutes).
  3. Be prepared to share your thoughts with the class!

# Deleting




- **Input**: List  $L$ ; key  $k$  you want to delete (e.g., 4)
- **Output**: void

# Deleting



- **Input**: List  $L$ ; key  $k$  you want to delete (e.g., 4)
- **Output**: void

 Respond at **PollEv.com/lauraburdick889**

 Text **LAURABURDICK889** to **37607** once to join, then **A or B**

On average, is it more efficient to **delete** an item in an array or a linked list?

- A. Deleting an array element is more efficient.
- B. Deleting a linked list element is more efficient.
- C. Both are the same efficiency.
- D. It depends.

## 1-minute Pause

- Stop and review the material we've covered so far (slides, notes, code, etc.).
- Is there anything that you need clarified?

