CSC 236 A08: A Simple Introduction to Linked Lists

Enter your name:

Some rules and procedures THAT YOU NEED TO READ (at least once):

- **This assignment is to be done individually, although you may discuss the work with the professor, the TAs, and even your classmates, making any copies of your work or allowing anyone else to copy your work is academic dishonesty and will be treated as such.**
- **Make a copy of this worksheet in your private Google account.**
- **Turn on sharing by link. Do not make it publically accessible by web, only by link.**
- **Share your copy with pearcej@gmail.com and no one else. Be sure you don't share your link with anyone else because this is academic dishonesty.**
- **Change the usernames in the document title as directed. (Your username is your Berea username--mine is pearcej.)**
- **Submit the sharing address (not the edit address) of your completed activity into Moodle.**

Read A Simple Introduction To Data Structures: Part One – Linked Lists and answer the following questions.

1. Describe what the author says a data structure is and why you need to learn to use them effectively.

2. Identify all of the linked-list terminology in the reading (eg. pointer, node, linked list, etc) For each of these,write the author's definition and explain what the author indicates is the main purpose of this concept

-
-
-
-

3. Describe when and why the author says you **should** use a linked list instead of other data structures.

4. Describe when and why the author says you should use **NOT** a linked list over and should choose another data structure.

Linked Lists are very, very different from both Python lists and C++ vectors because of the underlying memory usage!

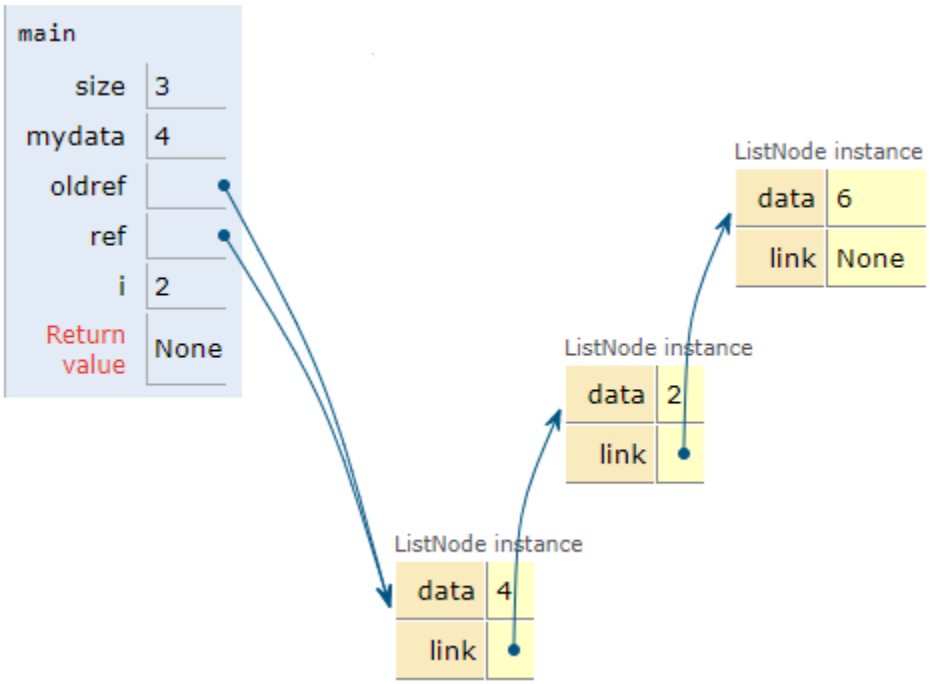
The Python list is implemented as a dynamically allocated array. This means the memory allocated to it must be contiguous, and sometimes during resizing, the entire list may need to move to a larger area of memory. We have learned that C++ vectors have many similarities to Python lists.

The **linked list** data structure is not like a Python list or a C++ vector. The primary similarity to the Python list and C++ vectors because all are linear structures, but there are also very, very significant differences.

Linked List Visualization

Consider the following image which is a visual representation of a linked list with three nodes where each “ListNode” is an instance of a simple class.

Note that none of the ListNodes are variables because they have no names or identifiers, so the only way to access the information in the ListNode data fields via pointers.



Note: This was created using PythonTutor.com with code that I wrote.We will see more of this excellent tool below.

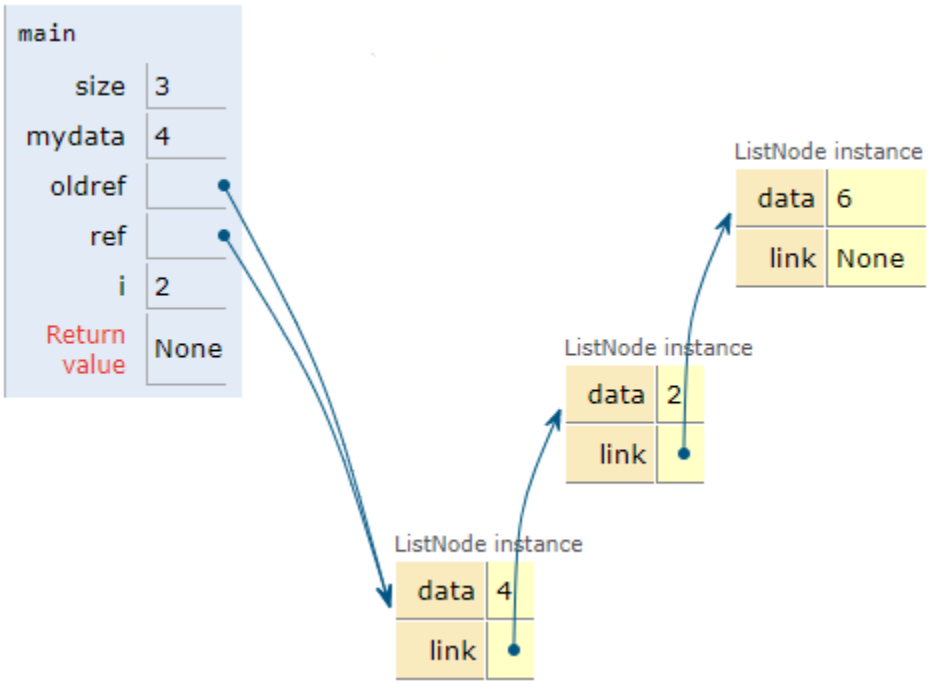
Using the terminology of the important features

5. Apply the key terms:
    a. Identify the variables serving as **head pointer(s)** in the above image.
    b. Identify the **head node** in the above image.
    c. Identify the **info field** and the **link or next field** of the head node in the above image. What are they called (their identifiers)?
    d. Identify **ALL nodes** in the above image. How many are there?
    e. Identify **ALL pointers** in the above image. How many are there? (Note that this visualizer represents pointers as arrows, since they "point" to a different memory location.

a.
b.
c.
d.
e.

Using a Linked-List Node in Python and C++

Next, you will visit the code which produced this image.



Exploring Linked Lists

6. (5 min) Visit both of the following perhaps on different laptops and step through the code:
   - Python Linked List Code
   - C++ Linked List Code

   a. Identify differences in the LinkNode class in each language version.

   b. Identify the lines in C++ Linked List Code which makes the arrows appear in the C++ version of the visualization. These are variables which contain memory addresses. In C++ these are called **pointer variables**. Write the C++ line(s) of code which make the arrows in the C++ version of the visualization.

   c. Identify the Python code in Python Linked List Code which makes the arrows appear in the Python version of the visualization. In Python these arrows are called **references** (and **every Python** variable is actually stored using a reference to it), which makes for a great deal of syntax simplification in Python. Write the Python line(s) of code which make the arrows in the Python version of the visualization.

7. Queues can be implemented in linked lists and doing so has some advantages for memory. Explore this Linked List Queue Visualization. Then discuss the advantages a linked-list implementation may have over an array implementation.

8. What is the key idea you learned in this assignment about linked lists?

Integrity statement

Please briefly describe all help you received and all help you gave to others in completing this assignment.

To submit

- **Turn on link sharing, but do not make it public on the web.**
- **Share your copy with pearcej@gmail.com. Be sure you do NOT share it with anyone else.**
- **Change the usernames in the document title as directed. (Your username is your Berea username--mine is pearcej.)**
- **Submit the sharing address (not the edit address) of your completed activity into Moodle.**
- **Turn on the ability to comment.**