**CS469 Data Structures and Algorithms**

**HOS02: Linked Lists and Selection Sort**

04/05/2023 Reviewed by Christopher Sharp

01/12/2024 Reviewed by Anh Nguyen

09/29/2024 Reviewed by Shahid Khan

School of Technology and Computing (STC) @City University of Seattle (CityU)

**Before You Start**

* The document’s examples are written in Python. If you don’t know Python programming language, please finish the Python tutorial in Module00 folder before you start the assignment.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Understand Linked list and the selected sort algorithm
* Implement the selected sort algorithm in Python

**Resources**

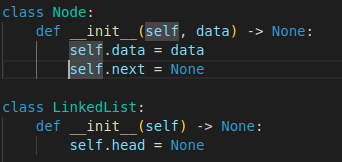
* Bhargava, Aditya. (2016). *Grokking Algorithms.* Manning Publications.
* Geeksforgeeks. (2021). Selected sort. Retrieved from: <https://www.geeksforgeeks.org/selection-sort/>
* TutorialPoint. (n, d). *Python – linked list.* Retrieved from: <https://www.tutorialspoint.com/python_data_structure/python_linked_lists.htm>
* Algorithm visualization. Retrieved from: <https://visualgo.net/en>

1. **Linked List**

A Linked list is a data structure in which each element links to the next element. Each node in a linked list holds data and a pointer, the pointer points to the next node.

**Implementation:**

* Open Visual Studio Code, Select **File** -> **Open**, Open the HOS folder that you cloned from GitHub Classroom (i.e: hos01-courseName-YourUserName)
* In the Module02 folder, create a new file called *linkedList.py*
* Type the following code in the***linkedList.py***file

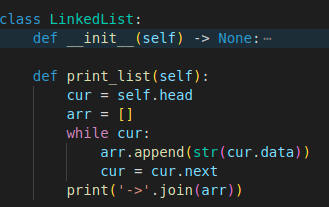


The code implements a linked list Node class, which is the smallest element of a linked list. Now let's implement the common operations!

* 1. Print

Before starting the following section, let's implement a helper function to print the current linked list.

* Add print\_list() method to class LinkedList. This will help print out all the nodes in the LinkedList by traversing from the head of the list.

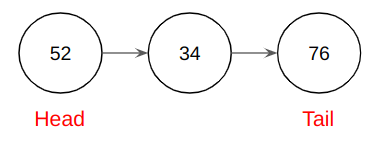


* 1. Insert

In this section, we introduce how to insert a node at the head of a Linked List by:

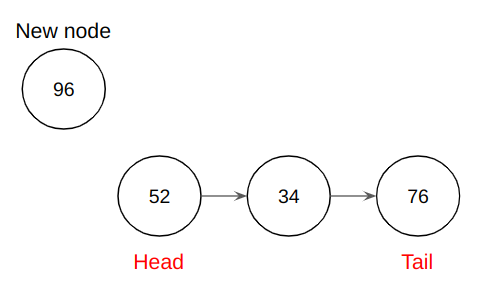
* Create a new node using the given data,
* Rewrite the new node’s pointer.
* Change the list’s head to the new node.

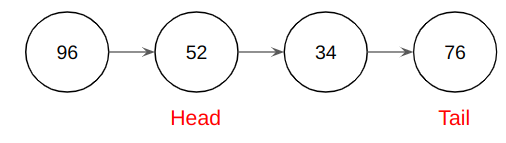
Take the following linked list as an example:

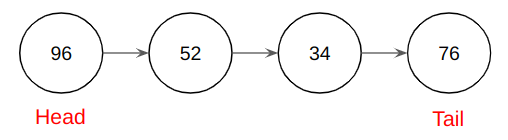


In the linked list, the head is node 52. It points to the next node 34, and the node 34 points to the tail node 76, while the node 76 points to None.

Let insert a node 96 at the head:



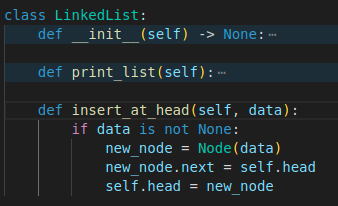




As you see, it created the node 96, then let the node 96’s pointer points to the node 52 and rewrite the head value.

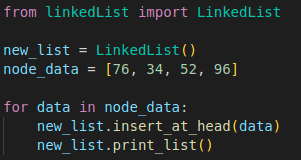
Next, write the insert function and create the example linked list.

* Add insert\_at\_head() function in LinkedList class:



Test the insertion function using the following code:

* Create a **new** file called ***linkedList\_operations.py***
* Type the following code in the***linkedList\_operations.py* file**



* Click run button on the right upper corner

Graphical user interface, text, application, chat or text message

Description automatically generated

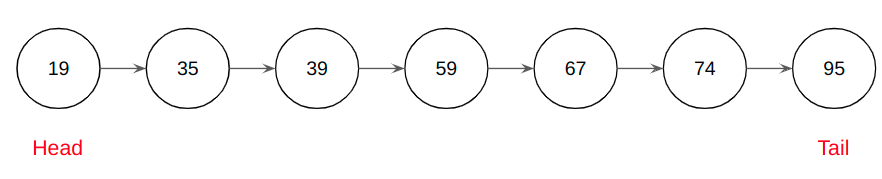
* The output should look like:

Text

Description automatically generated

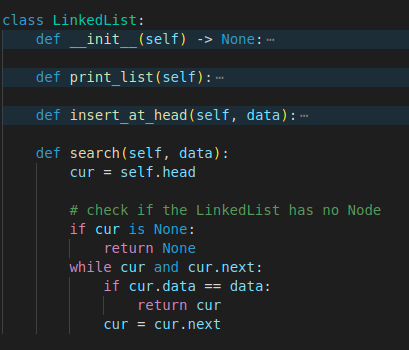
1.2 Search

The searching operation requires an iteration from head to the target node. If we want to find the node that contains data 67, the animation shows how the searching operation works:

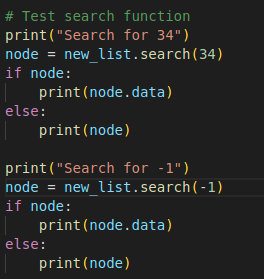


The time complexity of searching operation is O(n).

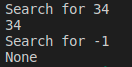
* Type the following code under class LinkedList in the***linkedList.py***file



* Type the following code in the***linkedList\_operations.py*** fileto test the deletion function



* Output should look like this:



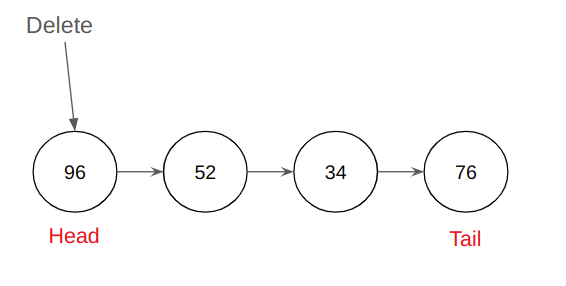
* 1. Delete

To operate the delete function, first you need to check if the deleted node is the head or not.

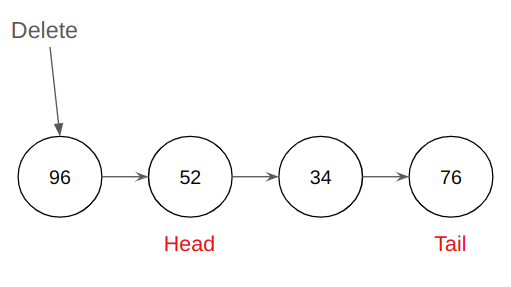
* If it is the head, you need to rewrite the head by the next node.
* Else, you need to find the previous node of the deleted node, then you can just rewrite the previous node’s next pointer to the next node of the deleted node.

The process is like the following animation:

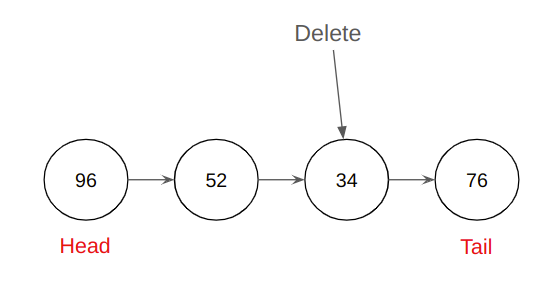
Case - Delete the head:



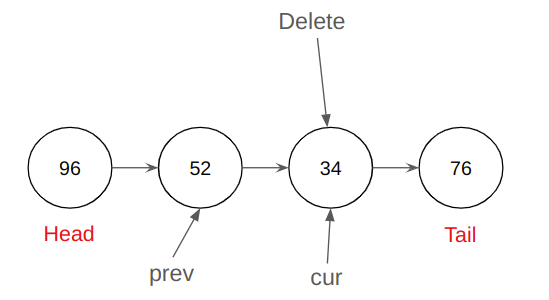
* Move the head cursor to the next node, in this case, node 52 shall be the next head. Node 96 is no longer considered to be in the LinkedList.



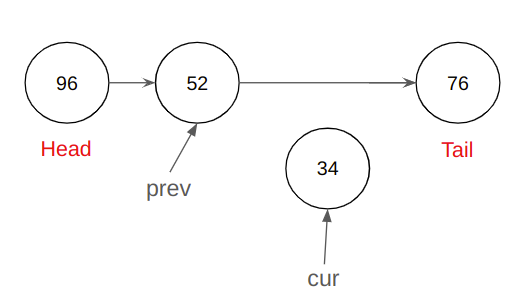
Case - Delete the third node:



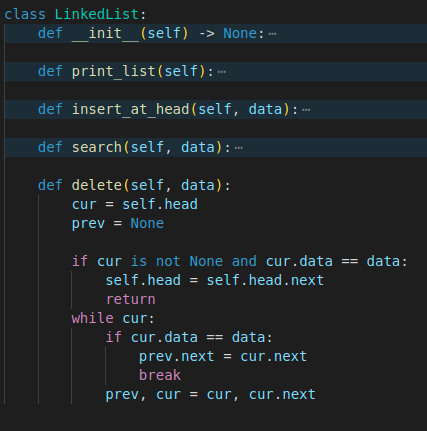
* Traverse from the head node, check if the node is the node we are looking for



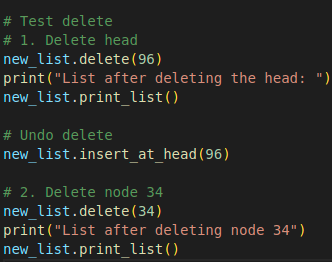
* + If false:
    - Assign prev to current node (cur)
    - Assign cur to the next node (cur.next)
  + If true:
    - Set prev.next = cur.next



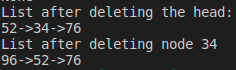
* + - Current node is no longer considered to be in the LinkedList
* Type the following code under LinkedList class in the***linkedList.py***file



* Type the following code in the***linkedList\_operations.py*** fileto test the deletion function

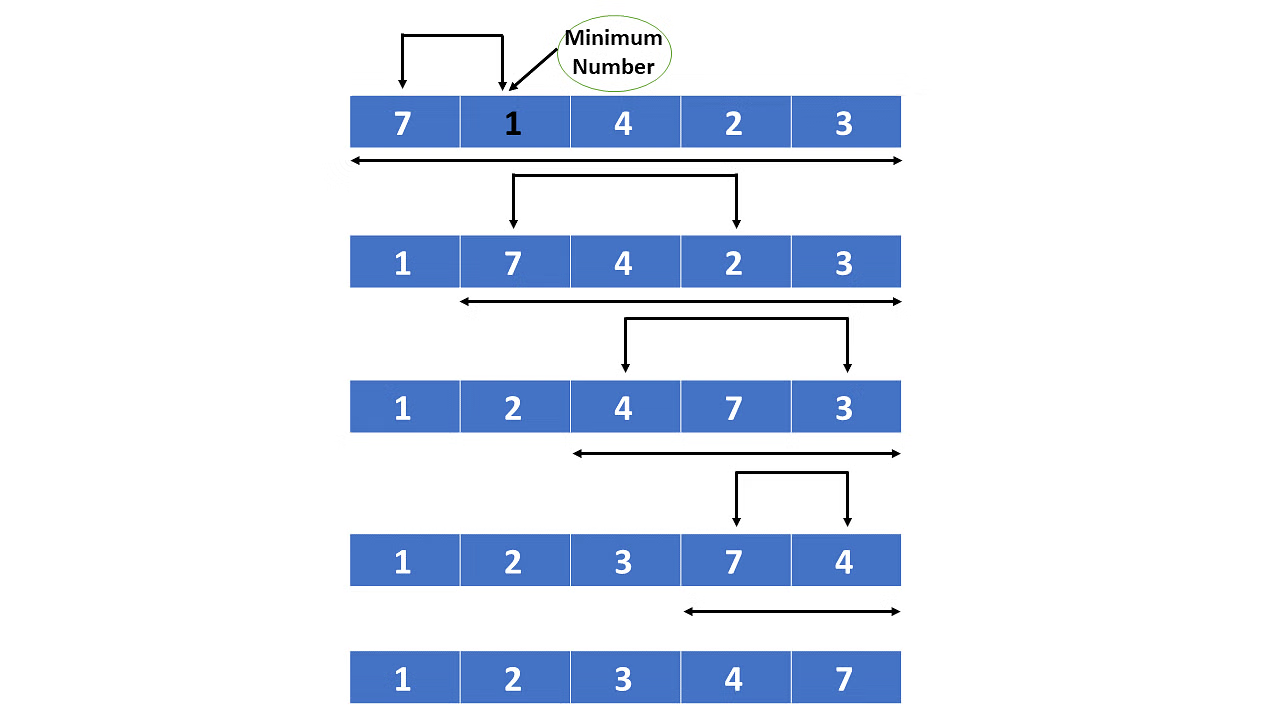


The output should look like:



1.3 **Selection Sort**

Selection sort algorithm repeatedly selects the smallest element from the unsorted portion of the list and swaps it with the first element of the unsorted part. This process is repeated for the remaining unsorted portion until the entire list is sorted.

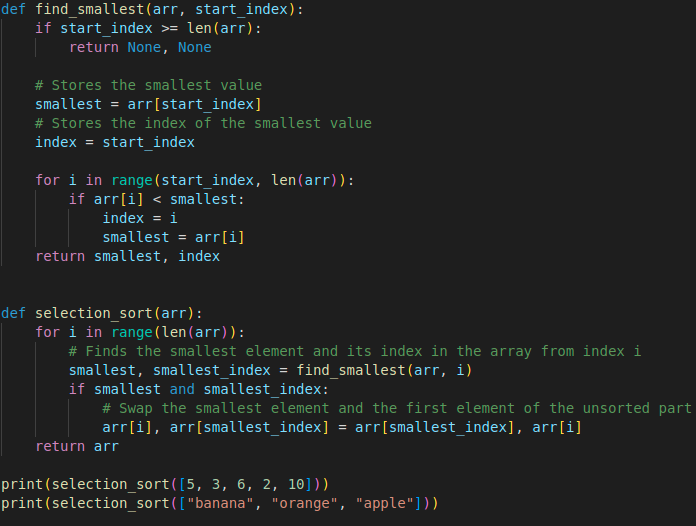


Selection Sort (simplilearn)

The time complexity of a selected sort algorithm is O (n2), and the space complexity is O (1).

**Implementation:**

* In the **Module02** folder, create a new file called *selectionSort.py*
* Type the following code in the *selectionSort.py* file



Output:

Graphical user interface, text

Description automatically generated

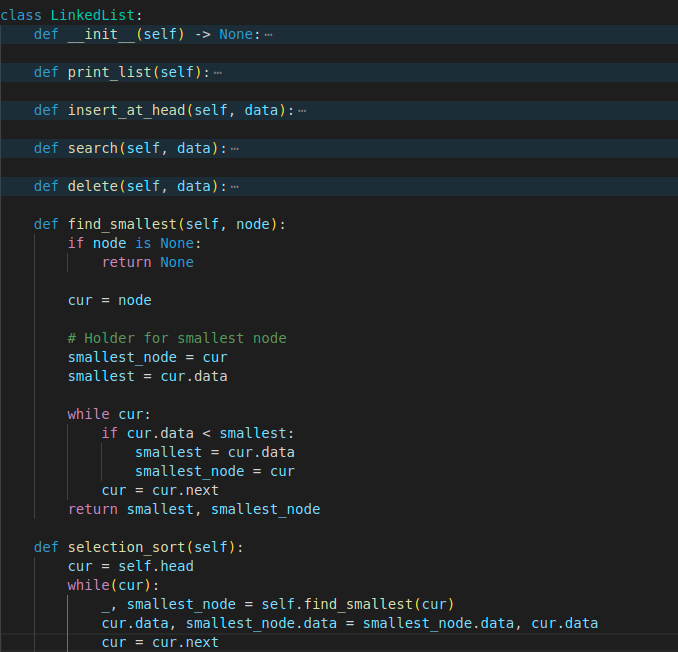
**Q: Could you modify this algorithm to return a descending array (from largest to smallest) ?**

Save your answer as a pdf file in the module folder.

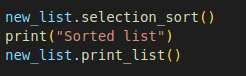
1. **Selection Sort on A Linked List**

You also can implement selected sort on a linked list:

* Type the following code under class LinkedList in the***linkedList.py*** file



* Type the following code in the***linkedList\_operations.py*** fileto test the deletion function



**Output:**

A picture containing text

Description automatically generated

**Push Your Work to GitHub**

Open terminal and make sure you’re in the repository folder. (i.e: hos02\_courseName\_GitHubUserName)

**Type the following command to upload your work**:

>>>> git add .

>>>> git commit -m “Submission for HOS02 - <Your name>”

>>>> git push origin master