What is the computational complexity of the methods in the implementation?

LinkedList class:

void addNode(T value, int index)

add a note to anywhere in the list

O(n)

void addNode(T value)

add a node to the end of the list

O(n)

void deleteNode(int index)

delete any node in the list

O(n)

void deleteNode()

delete the last node in the list

O(n)

Node<T>* getHead()

return the head of the list

O(1)

LineEditor class:

**n is the number of lines unless otherwise stated

void control()

receive user inputs and determine action

O(n), n is the length of the user input string

void insert(string text, int lineNum)

insert a line to anywhere in the line editor

O(n)

void insertEnd(string text)

insert a line at the end of the line editor

O(n)

void deleteLine(int lineNum)

delete any line in the line editor

O(n)

void deleteEnd()

delete the last line

O(n)

void print()

print the contents in the line editor

O(n)

bool checkString(string text, string target)

check if a string text contains the string target

O(mn), m is the length of the text, n is the length of the target

string trimString(string s)

trim the white spaces in the front or at the end of a string

O(n), n is the length of the string s

bool isNum(string text)

determine if the string text is a number

O(n), n is the length of the string text

void search(string text)

search if any line contains string text

O(n)

void edit(string text, int lineNum)

replace anyline with the string text

O(n)

void quit()

change the status of line editor to quit

O(1)

bool getQuit()

return if the line editor is closed

O(1)

Your thoughts on the use of linked lists for implementing a line editor. What are the advantages and disadvantages?

Advantages:

1. Inserting and deleting lines in the middle is quick and easy to implement. All we need to do is to change the pointers and delete the line if necessary.
2. Traversing through the lines is straightforward. It is the same as traversing through the nodes in the linked list.

Disadvantages:

1. Cannot directly access the lines. To get to a specific line, except for the first line (head), we always need to start from the first line (head) and traverse to that line.

What did you learn from this assignment and what would you do differently if you had to start over?

From this assignment, I learnt that simple data structures can be used to implement real-world applications, like the line editor in this case. Choosing the most suitable data structure for an application is very important because doing so makes the implementation easy and efficient.

If I had to start over, I will use the array-based data structures instead of linked lists to implement the line editor for several reasons:

1. Lines in the line editor have their line numbers, just line elements in the array are indexed.
2. Insert, delete, search and print will have similar computational complexity as linked lists, but insertEnd and edit are much quicker (O(1) vs O(n)) because we can directly access elements in the array with their indexes.