

Andrew Sanchez

1) Time complexity of methods:

- `search(string value)`: $O(n)$, despite having additional comparisons when checking each string, the complexity doesn't increase since a line in the document can't be longer than 80 characters.
- `addNode(string value)`: $O(1)$, since it adds directly to the end of the list, the complexity is constant
- `addNode(int index, string value)`: $O(n)$, specifying the index, requires iterating through the list until the index is reached. And since the size of the list can constantly increase, there's no limit to what the index could potentially be.
- `editNode(int index, string value)`: $O(n)$, same reasons as `addNode(int index, string value)`
- `deleteNode(int index)`: $O(n)$, same reason as edit and `addNode(int index, string value)`
- `printList()`: $O(n)$, since it needs to go through the entire list, printing every line.

2) Using a Linked List for a line editor works well over an array. While an array may be easier to edit lines with, a linked list is better at deleting and inserting lines since pointers are very flexible. Deleting a line from an array would require every element after the deleted element to be shifted over, making best and worst case $O(n)$.

3) I learned that when taking string inputs, `std::cin` isn't that great. Using `getline()` is much better. If I could start over, I would probably have made my main much smaller.