

Homework 8: Recursion and Nodes

Due 11:59pm Wednesday, November 19, 2025

CSCI 60

Krehbiel

Overview. This week's task is to implement four recursive functions, two of which relate to linked lists. Functionality is not enough; all your functions must be recursive. The first function should compute the n th value of a recursively defined sequence, the second is a recursive version of a function implemented for your midterm, the third recursively constructs a linked list from an array, and the fourth recursively deletes the dynamic memory associated with a linked list.

Implement all functions in `node.cpp`, the only file you should upload. Don't modify the corresponding four function signatures provided for you, and as always, do not add any libraries. A test file `main.cpp` is provided for you along with the `node.h` interface and corresponding inline implementations. As always, you should test as you go, which probably means commenting out most if not all of my code and writing your own test cases as you implement the different functions. It's your responsibility to use the main program effectively; you won't upload your test code. To test, compile as follows:

```
g++ main.cpp node.cpp -std=c++11
```

Function 1. Write a function to recursively compute the n th item in the following sequence: $a_0 = 1, a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 5, a_5 = 7, a_6 = 10, a_7 = 13, \dots$, where $a_n = a_{n-2} + n - 1$ for $n \geq 2$. Your function must be recursive.

Function 2. Write a function to recursively reverse the order of the elements in an array of integers. The function will be called on an array, specifying the first and last index of the array to reverse. Your array reversal should be in-place, meaning you should not allocate any additional arrays on the stack or on the heap. Your function must be recursive.

Function 3. Write a function to recursively construct a linked list with contents as specified by an array with specified length. Assume the initial recursive call is made with both head and tail set to null. When it returns, the head pointer should be point to a node with data `a[0]` that links to a node with data `a[1]`, and so on, with the tail pointer pointing to a node with data `a[n-1]`. Your function must be recursive.

Function 4. Write a function to recursively delete the memory for all nodes in a dynamically allocated linked list with head pointing to the first and tail pointing to the last node. The head and tail pointers should be null when the function is complete, and no memory should be orphaned. Your function must be recursive.