CS 136 Fall 2021

Tutorial 09 – Linked Data Structures

November 23, 2021
Today’s Topics

• Section 11
• Linked Data Structures
Section 11

Who has read it?

Any questions?

What I think is important:

Linked Data structure: container (“entry point”) + node (list) / nodes (tree)

When traversing or mutating linked data structures, consider:
• Container has nodes vs. container has no node (e.g., first insert, last remove)
• Changing data structure at the container vs. in the middle vs. at the end

Make sure to keep cached data consistent.

When using recursive traversal, the majority of functionality happens on nodes, since recursion happens on nodes and not on the container.

Whenever you have to mutate a linked data structure, MAKE A DRAWING.
Whenever you have to debug a linked data structure, MAKE A DRAWING.
Exercise – container-less linked list

- Implement the module llnode: the module forms a linked list without using a “container”-structure like llist.
  - To make your life easier, implement the functions in order
  - You must change the signature of one function due to the fact that llnode is container-less.
  - It is your decision, if you want to implement destroy, print, and insert using an iterative or a recursive approach. Why not mix it up a bit!
  - For find_max, try both approaches.
Exercise – sorted linked list

• Finish the implementation of `llnode`: the module forms a doubly-linked list without using a “container”-structure like `llist`.
  • Implement `insertion_sort_values`: the function sorts a linked list by sorting the value, i.e., mutating the values in each node.
  • Implement `insertion_sort_nodes`: the function sorts a linked list by sorting the nodes, i.e., changing the order of nodes in the linked list.
  • Try to find conceptual analogies between the array-notation of insertion sort and the linked-list approach:
    • What are `int i = 1` and `int j = i`?
    • What are you iterating over, i.e., what does `++i` and `--j` mean on a linked list?
    • What are `a[j]` and `a[j - 1]`?
    • How do you swap nodes in a linked list? (DRAW IT OUT!)
  • You must change the signature of one function due to the fact that `llnode` is container-less.
Exercise – sorted linked list

• Implement `insertion_sort_values`: the function sorts a linked list by sorting the value, i.e., mutating the values in each node.

• Implement `insertion_sort_nodes`: the function sorts a linked list by sorting the nodes, i.e., changing the order of nodes in the linked list.