

Tutorial (Advanced Programming) Worksheet 8:

Assignment 1: Lists

List are among the most frequently used algorithmic patterns in computer science. Instead of using lists from the `std` library, we develop our own modest list library.

For this assignment, only integer values should be stored to the list. Write a struct or class which offers the following features:

- *push_front*: Push (insert) an element before the first element.
- *pop_front*: Pop (remove) an element before the first element.
- *push_back*: Push (insert) an element after the last element.
- *pop_back*: Pop (remove) an element after the last element.
- *output()*: Output integers stored in list.

There are plenty of ways to implement this list. Consider pros and cons when implementing it and make a sketch of the heap.

Assignment 2: Binary tree

Binary trees can be used for storing and searching words in a dictionary.

Develop a program which allows inserting a tuple (*int*, *string*) assigning a string to an integer.

The datastructure which you have to program has to handle each request in $O(\ln(n))$ operations.

- *insert*: Insert an element (*int*, *string*). If another element already exists, it should be overwritten.
- *find*: Find the string associated to integer and return string. In case that no element exists, an empty string should be returned.

Homework assignment 1: Binary tree

Implement a *delete* feature for the binary tree.

Questions:

Answer the following questions:

- Assignment 1: What is the complexity of the different methods?
- Assignment 1: Can you imagine ways to reduce the complexity?
- Which ways do you know to detect memory leaks?
- Cache efficiency can be improved by a customary heap-management. How does this look like?
- Make yourself known to tree-rebalancing. What are pros and cons as well as aims of tree-rebalancing.