

1 General remarks

This week is about

- heaps
- HeapSort.

See the script for Week 9 on Canvas.

We inspect Java-code. For the lab you don't need to run the code (but later it should be really helpful to run examples with it, say at home).

2 Understanding the implementation

Look into `Heap.java` (on Canvas, in module "Programs for labs") and understand the code.

Q1 How can we construct Heap-objects? After construction, do we already have a binary heap?

Q2 What is the purpose of the function `leaf`? Create some examples yourself, to check whether the computation is actually correct.

Q3 For a node of the binary heap (as a tree), or, within the program, for an index $0 \leq i < n$, if the node is not a leaf, is it guaranteed that there is a left child?

Q4 Similarly, is it possible that there is no right child? If yes, for which nodes can this happen?

Q5 What is `heap_property` computing? Why is it correct?

Q6 Which function changes n ? What is the final value of n . after executing this function?

Q7 Describe in a few sentences what the function `heapify` is doing.

Q8 Why is `build_heap` starting its action with $i = \text{fl} - 1$?

Q9 Describe in a few sentences how `sort` is working.

3 Running an example

Consider the sequence

20, 0, 26, 33, 4, 15, 7, 27, 2, 1, 33, 25, 0, 30, 21, 10, 5.

Q10 Build a binary heap from that sequence (on paper).

Q11 Now run HeapSort on that sequence, computing one after another all the binary heaps which are created on the way (as we had it in the example from the lecture).