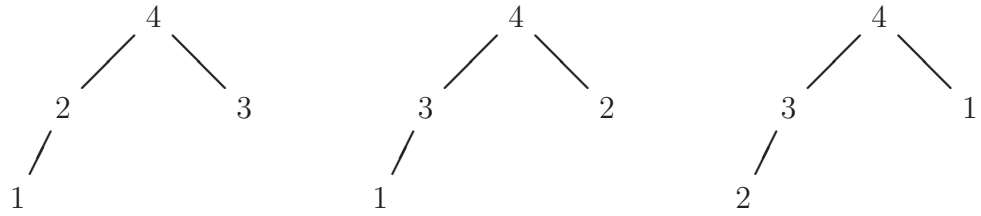
ALGORITMIEK: some solutions to exercise class 12

Problem 1.

- Values 1–4: Value 4 should be the root of the heap. We need to partition the remaining values 1-3 over the two subtrees: two values in the left subtree, and one value in the right subtree (because the tree must be complete). There are three possible partitionings: {1,2} - {3}, {1,3} - {2}, {2,3} - {1}. Each of these partitionings is fine.

  Given a partitioning, there is one way to store the two elements in the left subtree (the larger one should be the root) and the only element in the right subtree.

  We thus have three possible heaps:



- Values 1–5: Value 5 should be the root of the heap. We need to partition the remaining values 1-4 over the two subtrees: three values in the left subtree, and one value in the right subtree (because the tree must be complete). There are four possible partitionings: {1,2,3} - {4}, {1,2,4} - {3}, {1,3,4} - {2}, {2,3,4} - {1}. Each of these partitionings is fine.

  Given a partitioning, there are two ways to store the three elements in the left subtree (the largest one should be the root, one of the other two should be the left child, and the remaining one the right child) and the only element in the right subtree.

  This yields $4 \times 2 = 8$ possible heaps in total.

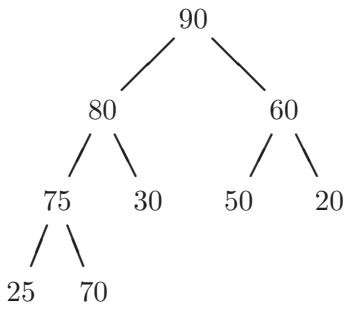
- Values 1–6: Value 6 should be the root of the heap. We need to partition the remaining values 1-5 over the two subtrees: three values in the left subtree, and two value in the right subtree (because the tree must be complete). In other words: we need to choose three out of five values for the left subtree. The other two values then go to the right subtree. There are $\binom{5}{3} = 10$ possible choices / partitionings: {1,2,3} - {4,5}, {1,2,4} - {3,5}, {1,2,5} - {3,4}, {1,3,4} - {2,5}, {1,3,5} - {2,4}, {1,4,5} - {2,3}, {2,3,4} - {1,5}, {2,3,5} - {1,4}, {2,4,5} - {1,3}, {3,4,5} - {1,2}. Each of these partitionings is fine.

  Given a partitioning, there are two ways to store the three elements in the left subtree (the largest one should be the root, one of the other two should be the left child, and the remaining one the right child) and the two elements in the right subtree (the larger one should be the root).

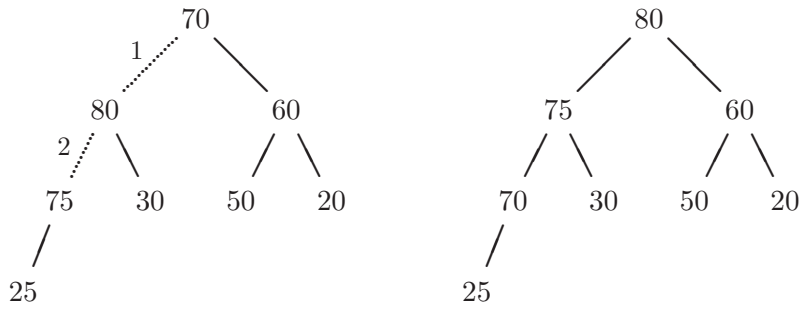
  This yields $10 \times 2 = 20$ possible heaps in total.

The careful reader should see a dynamic programming approach to compute the number of possible heaps containing values 1–n.

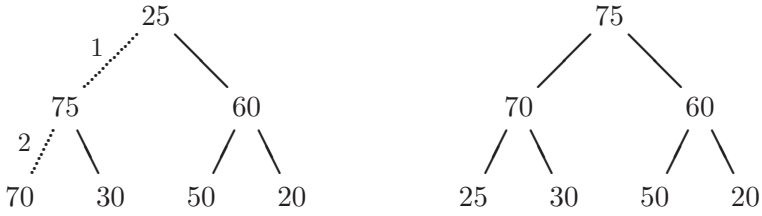
Problem 3. The sequence 50, 70, 60, 75, 30, 90, 20, 25, 80 corresponds to the tree from Problem 2. We have seen there, that heapify yields



which corresponds to the sequence 90, 80, 60, 75, 30, 50, 20, 25, 70.
We apply deletemax, that is: we remove 90 from the root of the tree, move the last leaf 70 towards the root (yielding the left tree below), and subsequently sift it down the tree (yielding the right tree below).

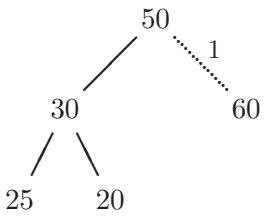


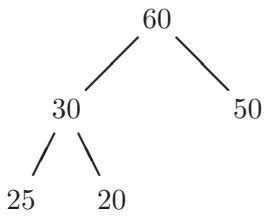
We again apply deletemax, that is: we remove 80 from the root of the tree, move the last leaf 25 towards the root (yielding the left tree below), and subsequently sift it down the tree (yielding the right tree below).



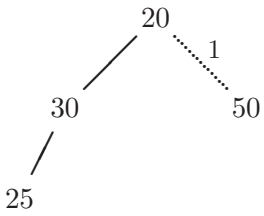
We again apply deletemax, that is: we remove 75 from the root of the tree, move the last leaf 20 towards the root (yielding the left tree below), and subsequently sift it down the tree (yielding the right tree below).

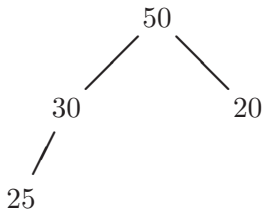
We again apply deletemax, that is: we remove 70 from the root of the tree, move the last leaf 50 towards the root (yielding the left tree below), and subsequently sift it down the tree (yielding the right tree below).



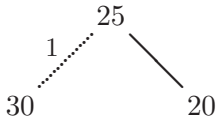


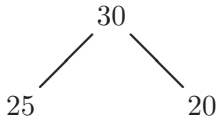
We again apply deletemax, that is: we remove 60 from the root of the tree, move the last leaf 20 towards the root (yielding the left tree below), and subsequently sift it down the tree (yielding the right tree below).



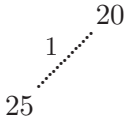


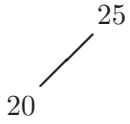
We again apply deletemax, that is: we remove 50 from the root of the tree, move the last leaf 25 towards the root (yielding the left tree below), and subsequently sift it down the tree (yielding the right tree below).





We again apply deletemax, that is: we remove 30 from the root of the tree, move the last leaf 20 towards the root (yielding the left tree below), and subsequently sift it down the tree (yielding the right tree below).





We again apply deletemax, that is: we remove 25 from the root of the tree, move the last (only) leaf 20 towards the root (yielding a tree consisting of only one node 20), and we are done. The resulting, sorted sequence of numbers is 20, 25, 30, 50, 60, 70, 75, 80, 90.

Problem 6. (Travelling Salesman Problem) This problem has been worked out in the slides of lecture 12.