Write your solutions in this sheet below the problem statement. Mark clearly your solutions by corresponding letters A, B, C, D and separate visually each solution from the other ones by empty space or by a line. Use the other side of the sheet or ask for an additional blank sheeet if necessary.

Each problem 1. – 4. is worth 0 – 4 points, each answer to a particular question A, B, C, D contributes at most 1 point to the total.

1. Four algorithms process a square matrix M of size $n \times n$. The indices of rows and columns of M start with 1. Eaxh of the algorithms process all metrix elements. Determine the asymptotic complexity of each algorithm.

A. Algorithm A processes each element of M in time proportional to $\log(n \times n)$.

B. Algorithm B processes each element in the row r in time $k \cdot r$, where k is a positive constant.

C. Algorithm C processes each element at position [r][c] (r and c are row and colum indices, respectively) in time $k \cdot r \cdot c$, where k is a positive constant.

D. Algorithm D processes each element in constant time. However, before processing the element, the algorithm checks the value of each element in the row in which the processed element is located. Each check takes constant time to complete.

2. Insert sort processes an array A with an even number of values, array length is N. The array contains elements of only two values, 10 and 20. The first value in the array is 10 and the values in the array strictly alternate: 10, 20, 10, 20, 10, etc.

A. Determine how many comparisons between two element values are performed when the length of the array is 8:

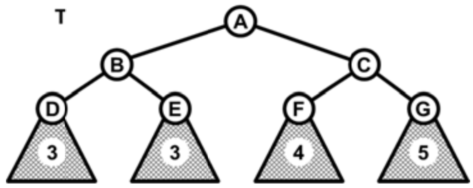
10 20 10 20 10 20 10 20

B. Determine how many comparisons between two element values are performed when the last element with value 10 is being inserted to its final position in array A. Solve the problem for the general value of $N \geq 4$.

C. Write a function $f(N)$ which returns the total number of comparisons made during the sort of array A. Suppose $N \geq 4$. Explain how did you derive $f(N)$.

D. Decide to which complexity class belongs $f(N)$. The available clasees are $\Theta(1)$, $\Theta(\log N)$, $\Theta(N)$, $\Theta(N \cdot \log N)$, $\Theta(N^2)$.

3. AVL tree T is in the picture. The four particular subtrees with roots in D, E, F, G are shown schematically and the value given in the subtrees represents the depth of the sbtree. The depth is the number of edges on the path from the subtree root to the deepest node in the subtree.

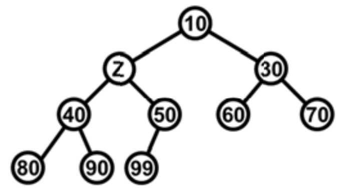


A. T is depicted immediatelyafter a new key (not depicted) was inserted in to the tree. To complete correctly the insert operation, a rotation hast to be applied in T. Draw the shape of T after this rotation and determine the type of the rotation (L, R, LR or RL).

B. Insert keys 10, 30, 20, 50, 40, in this order, into an originally empty AVL tree, using Insert operation. Draw the resulting AVL tree and determine the number and the type (L, R, LR, RL) of each rotation used in the process.

C. Delete the key 10 from the AVL tree built in C. Which rotation, if any, was applied in the deletion process?

D. Suppose that an AVL tree contains n^2 keys and that another n keys are inserted, one by one, into the tree using the Insert operation. Use the asymptotic notation (O/Θ/Ω) to specify the maximum number of simple (L or R) rotations which are performed doring the whole process. Explain your reasoning.



4. A binary heap is given in the picture.

A. Determine all possible integer values of element Z in the heap.

B. Substitute Z by its smallest possible value (calculated in the answer to question A) .

Next, use the Insert operation to insert values 55, 45, 35 into the heap, in the given order.
Draw the resulting heap.

C. Apply twice operation ExtractMin to the heap obtained in B. Draw the resulting heap.

D. Suppose that a binary heap H contains exactly $2n$ leaves. Operation ExtractMin is applied n times on H. What is the asymptotic complexity of the whole process? Explain your reasoning.