Problem 1. *Illustrate the operation of HOARE PARTITION on the array (No sorting required)*

$$A = \langle 13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11 \rangle$$

Problem 2. *Illustrate the operation of LOMUTO PARTITION on the array (No sorting required)*

$$A = \langle 13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11 \rangle$$

Problem 3. *For the set of* $\{1, 4, 5, 10, 16, 17, 21\}$ *of keys, draw binary search trees of heights 2, 3, 4, 5, and 6.*

Problem 4. *A concatenate operation takes two sets* S_1 *and* S_2*, where every key in S1 is smaller than any key in* S_2*, and merges them together. Give an algorithm to concatenate two binary search trees into one binary search tree. The worst-case running time should be* $O(h)$*, where h is the maximal height of the two trees.*

Problem 5. *Insert the following list in an* ***AVL*** *tree, start with an empty tree*

$$A = \langle 10, 20, 15, 25, 30, 16, 18, 5 \rangle$$

you always need to check if the tree is balanced after every new element insertion.

Problem 6. *Show that an n-element heap has height* $\lfloor log_2\, n \rfloor$
If you can not show it mathematically you can give an example to illustrate your answer

Problem 7. *Show that there are at most* $\lceil \frac{n}{2^{h+1}} \rceil$ *nodes of height h in any n-element heap.*

Problem 8.

$$A = \langle 5, 13, 2, 25, 7, 17 \rangle$$

- *Use the previous list to build a Red-Black tree, show steps of insertion by drawing a new tree for every step, use two different pen colors or shade the black node.*

Problem 9. *Illustrate the operation of HEAP-SORT on list A.*

$$A = \langle 5, 13, 2, 25, 7, 17 \rangle$$

Problem 10. *Use the following formula to argue the run-time of Max-Heap:*

$$\sum_{h=0}^{\lfloor log_2 n \rfloor} \left\lceil \frac{n}{2^{h+1}} \right\rceil * O(h)$$

Problem 11. *What are the minimum and maximum numbers of elements in a heap of height h? If you are performing Max-heap - on a complete binary tree- which node should you start with? support your answer with the correct formulas.*

Problem 12. *Where in a max-heap might the smallest element reside, assuming that all elements are distinct?*