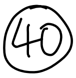
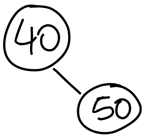
Mehmet Akif Şahin- 22203673

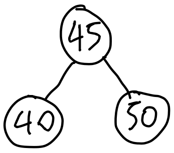
CS202- section 3 spring 2024

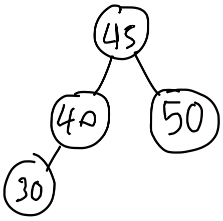
18 march 2024

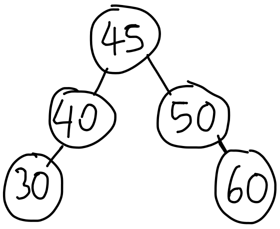
Insert 40, 50, 45, 30, 60, 55, 20, 35, 10, 25

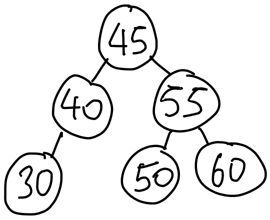


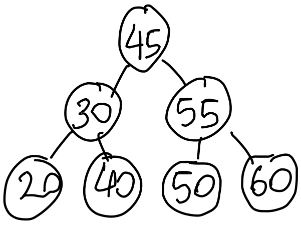


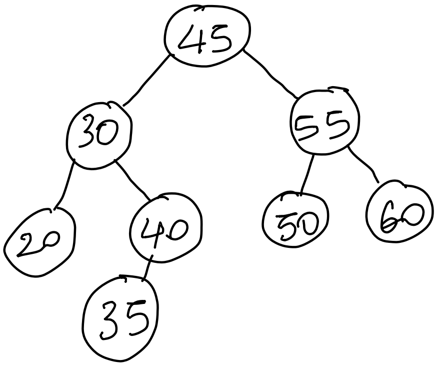


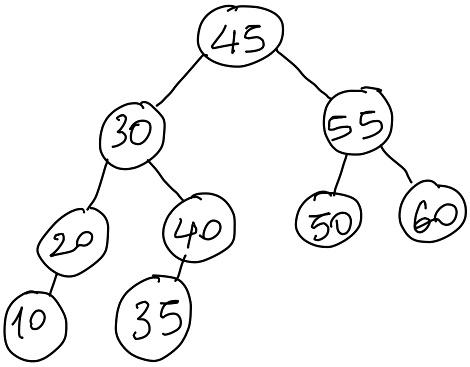


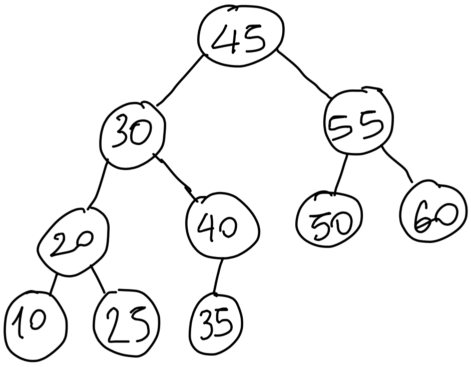






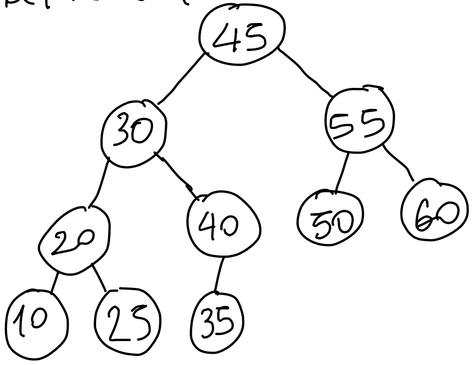


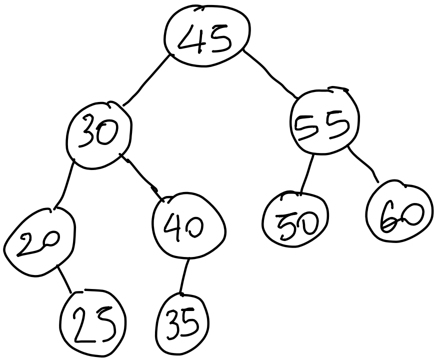


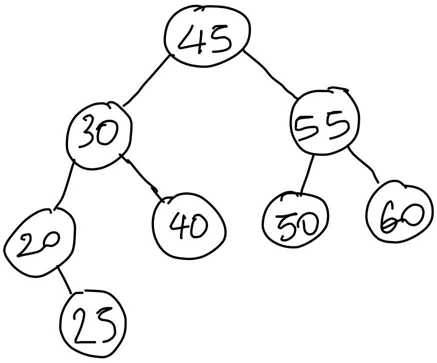


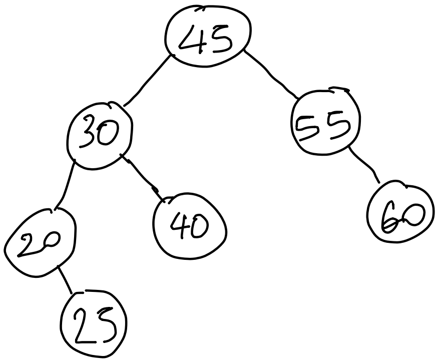
delete 10, 40, 50 i

before deletion

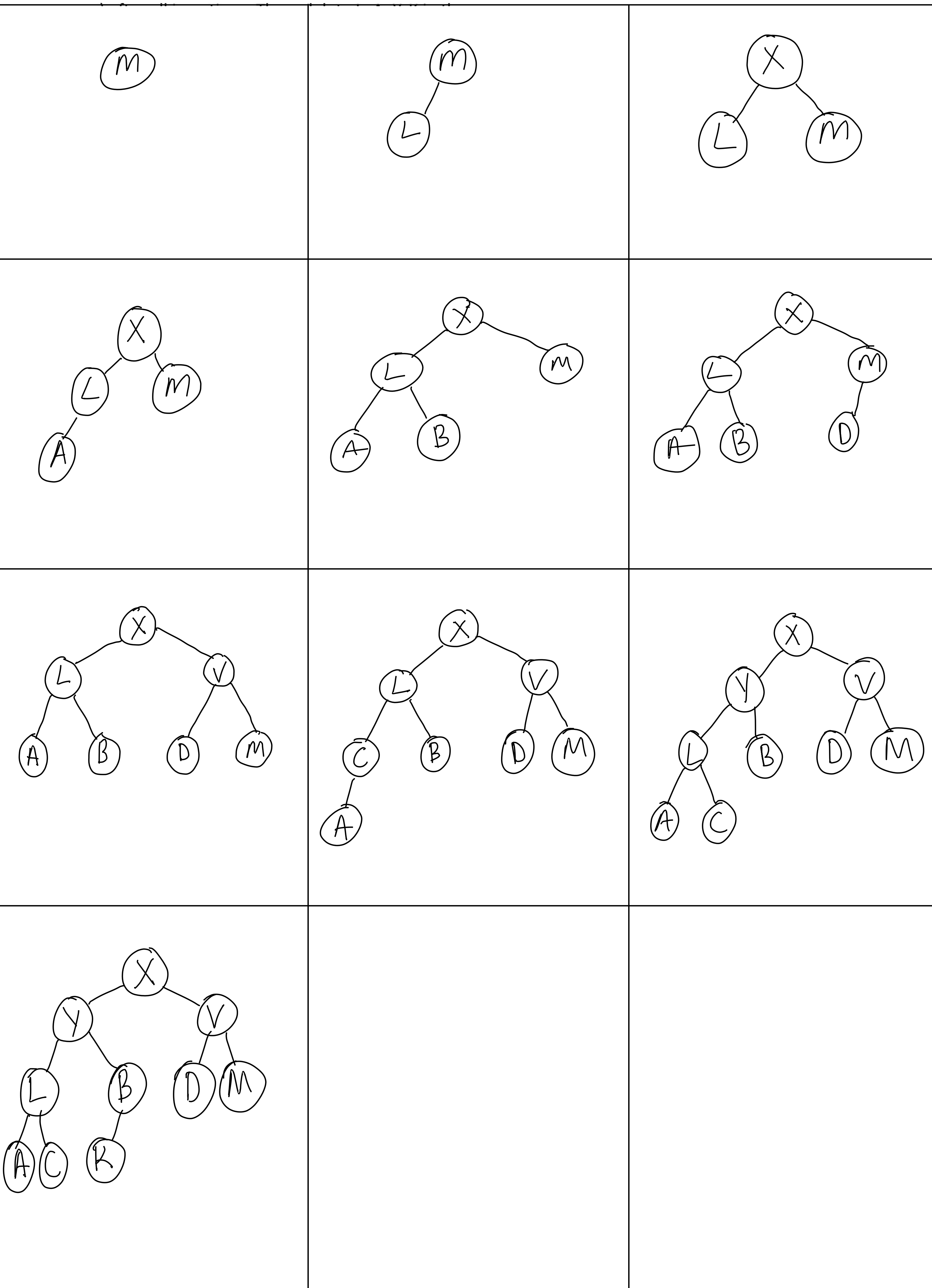






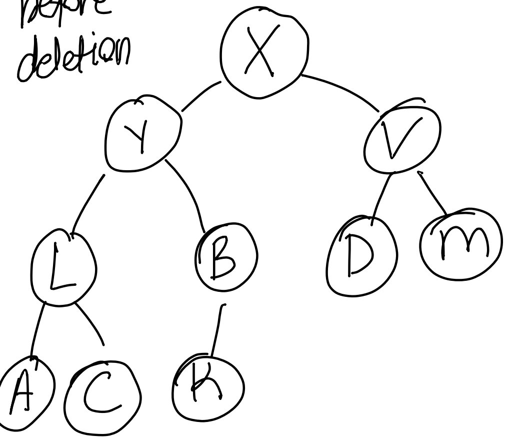


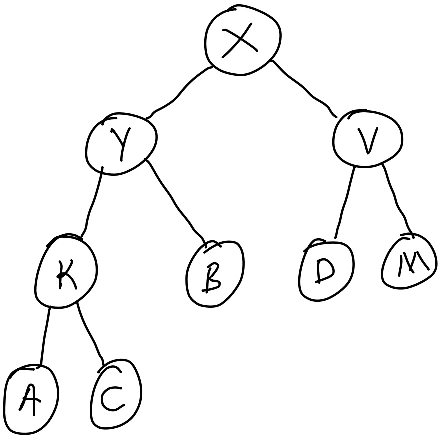
1 Insert M, L, X, A, B, D, V, C, Y, K, into an empty max-heap.

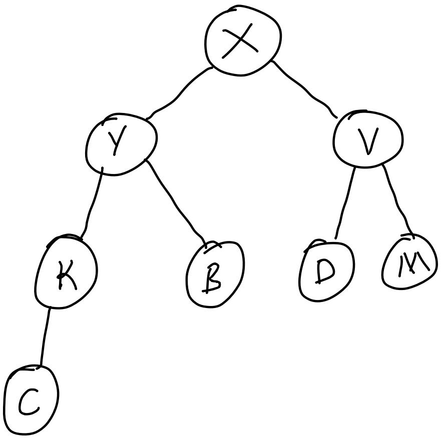


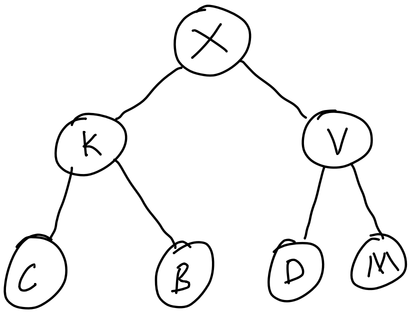
delete L, A, Y, K

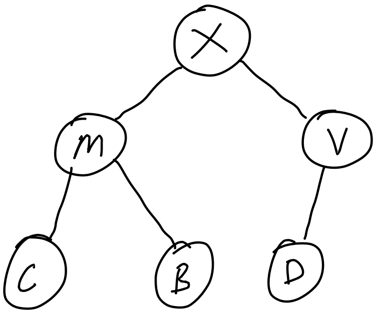
before deletion











(c) ***[5 points]*** You would like to store a set of numbers either in a max-heap or a sorted array. For the following applications, explain which data structure is better, or discuss if it does not matter which one is used. Answers without explanation will get 0 points.

(a) finding maximum element quickly

(b) finding minimum element quickly

(c) finding median of elements quickly

(d) deleting an element quickly

(e) forming the structure quickly

a) Max heap or sorted array can be used becauwe finding max element is O(1) in both.

b) sorted array should be used because finding min element in sorted array is O(1) (first element). In max heap min element is in leaves and complexity of traversing leaves in max heap can change implementation to implementation, in array implementation it is O(n).

c) sorted array should be used. Median can be found simply by looking at size/2 `th element, so it is O(1).

d) They both can be used. Deleting an element in heap reqires o(n) time because locating the element is O(n) even though deletion is O(log(n)). Deleting an element in array is similar. Locating element is O(log(n)) but deletion is O(n).

e) Both can be used. Forming heap from and array is O(n) but it is a very tight bound. Forming sorted array is basically sorting so it is O(n.log(n))