1. Suppose we have a list [5, 6, 3, 1, 2, 7, 9, 8]. Show the list at each stage when using:

a. Insertion sort:

- 5

- 5, 6

- 3, 5, 6

- 1, 3, 5, 6

- 1, 2, 3, 5, 6

- 1, 2, 3, 5, 6, 7

- 1, 2, 3, 5, 6, 7, 9

- 1, 2, 3, 5, 6, 7, 8, 9

b. Selection sort:

- 5, 6, 3, 1, 2, 7, 9, 8

- 1, 6, 3, 5, 2, 7, 9, 8

- 1, 2, 3, 5, 6, 7, 9, 8

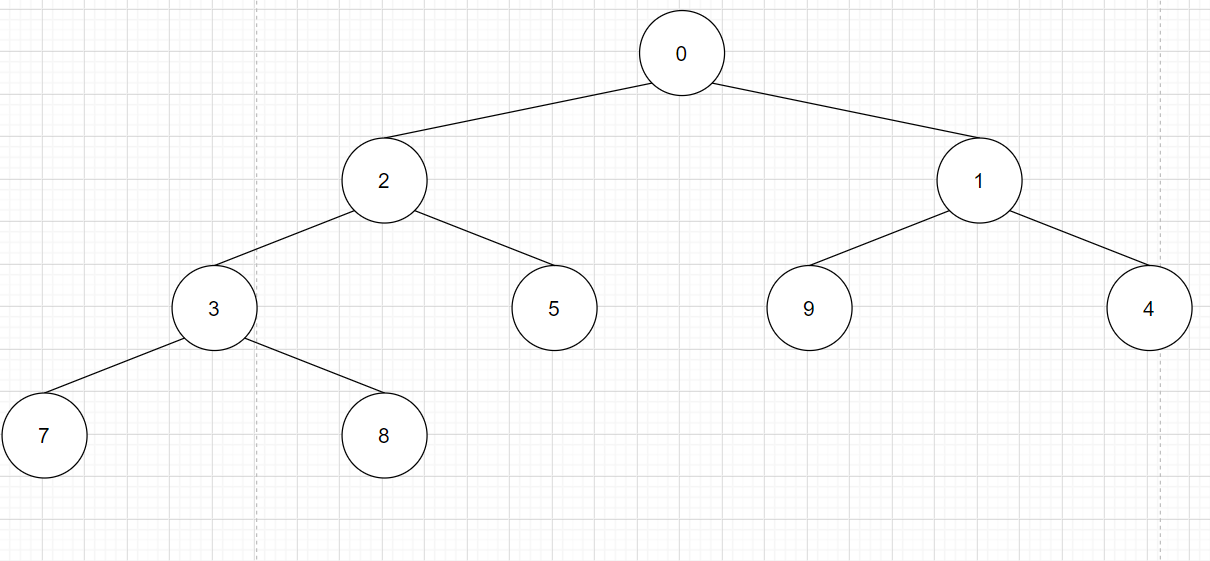
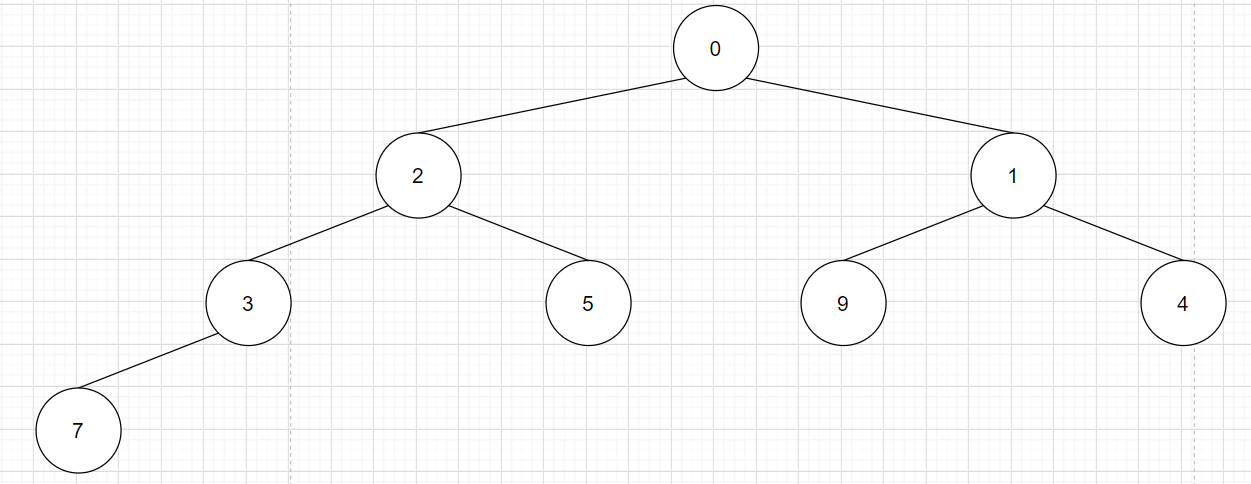
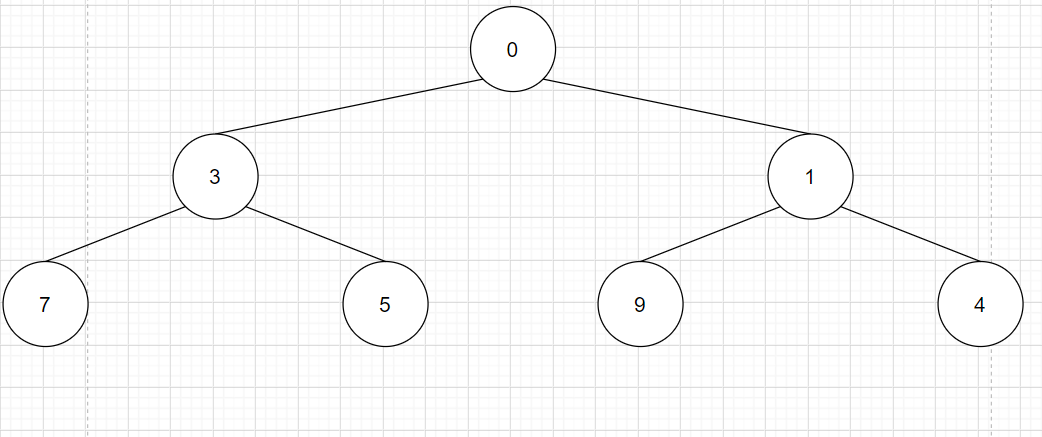
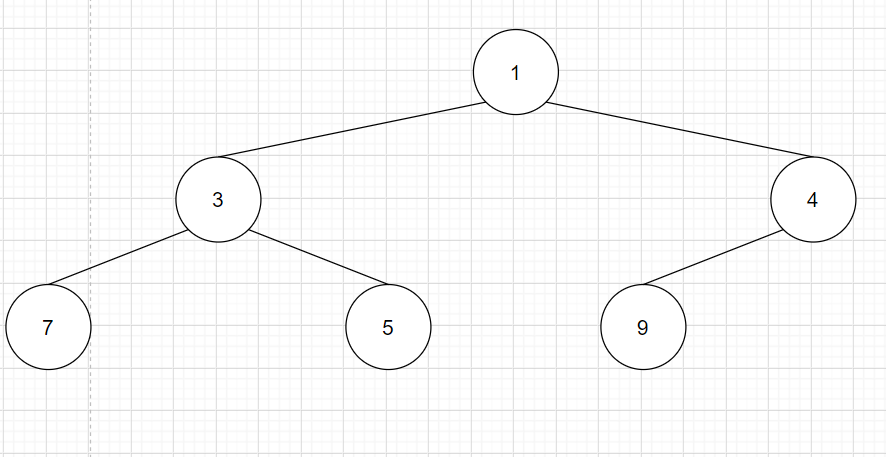
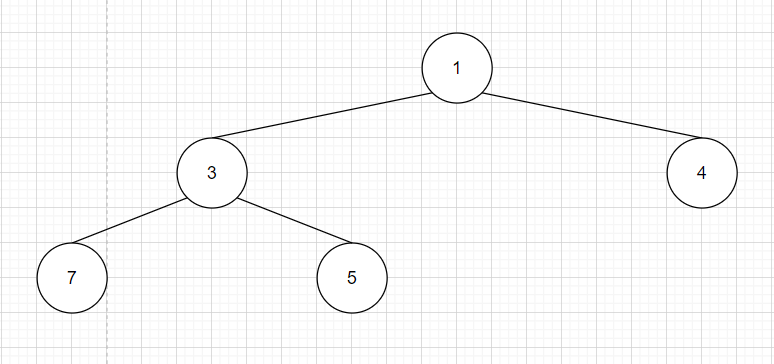
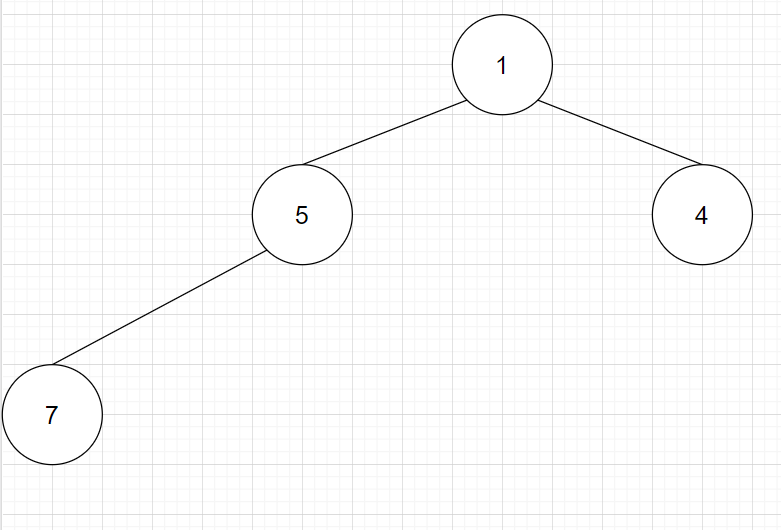
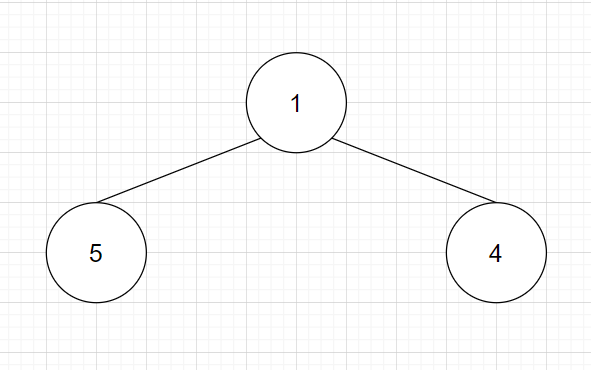
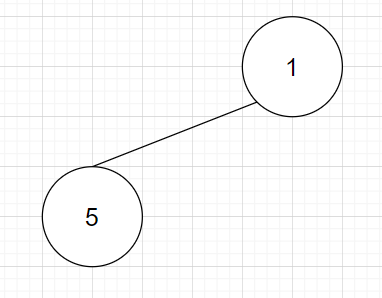
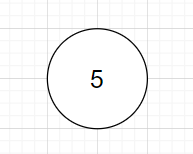
2. Give an example of a worst-case sequence with n elements for insertion-sort, and show that insertion-sort runs in O(n2) time on such a sequence.

Example Worst Case: [6, 5, 3, 2, 1]

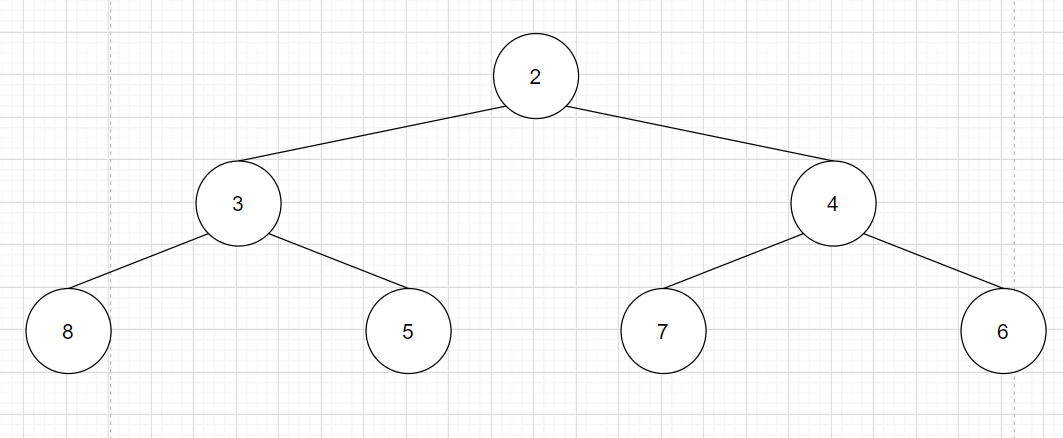
* 6 0
* 5, 6 n - 4
* 3, 5, 6 n - 3
* 2, 3, 5, 6 n - 2
* 1, 2, 3, 5, 6 n - 1

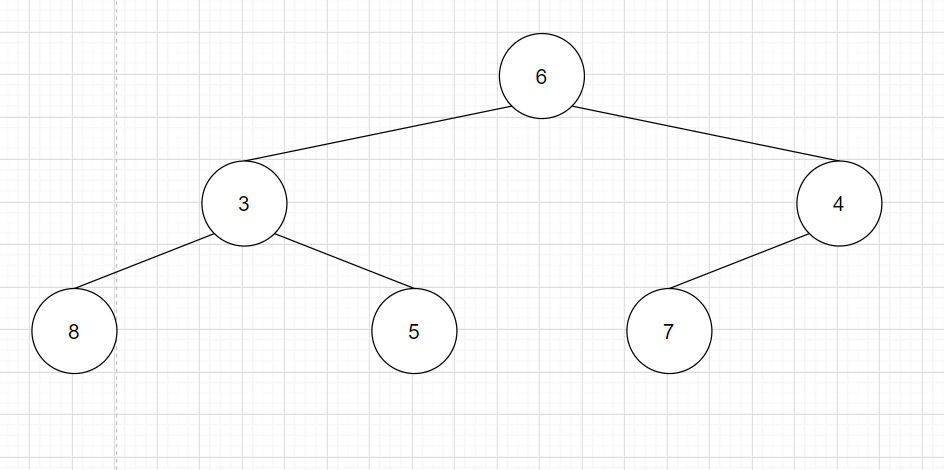
Time complexity: 0 + n - 4 + n - 3 + n - 2 + n – 1 = n(n-1) = n^2 – n = O(n2)

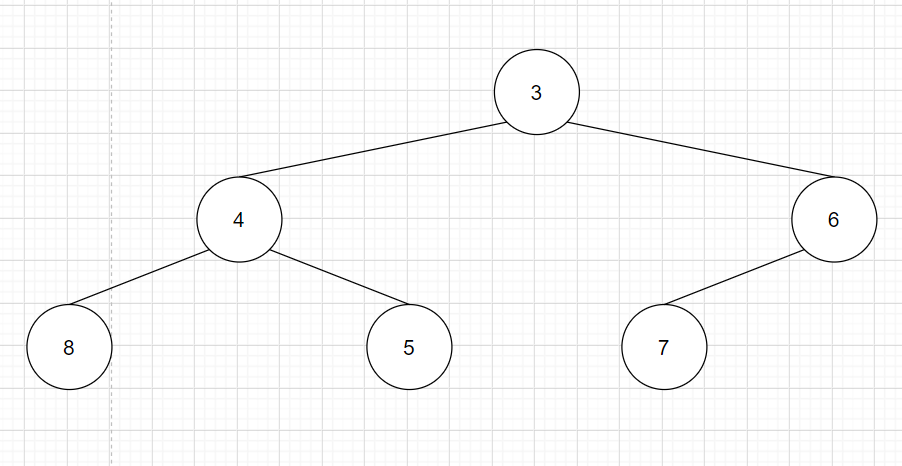
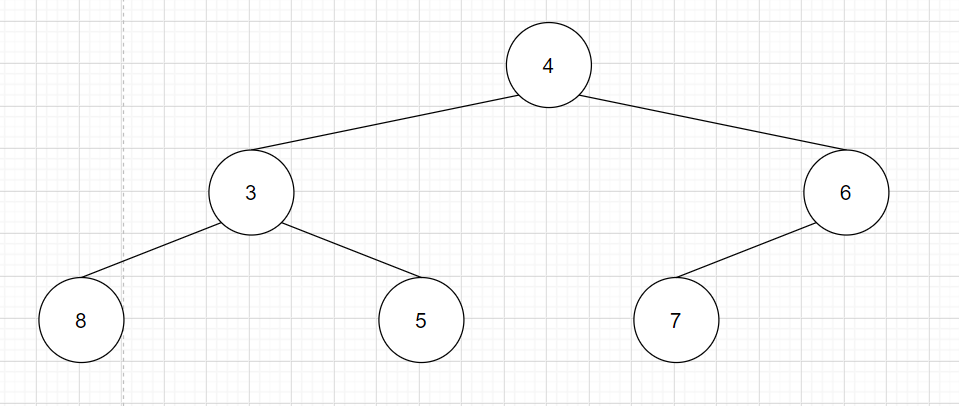
3. Show the state of an initially empty heap at each point as the following keys are added: 5, 1, 4, 7, 3, 9, 0, 2, 8.



4. Show all the steps of the algorithm for removing key 2 from the heap







5. At which positions of a heap might the third smallest key be stored? – Not at root

6. At which positions of a heap might the largest key be stored? – At leaves