LM Data Structures, Algorithms, and Databases (34140, 34141, 34139, 36989)

Exercise Sheet

Week 07 (Stack, Queue, Heaps)

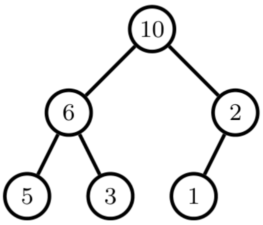
Q1. [Stack] Describe the output of the following series of stack operations: push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop()

Q2. [Queue] Describe the output for the following sequence of queue operations: enqueue(5), enqueue(3), dequeue(), enqueue(2), enqueue(8), dequeue(), dequeue(), enqueue(9), enqueue(1), dequeue(), enqueue(7), enqueue(6), dequeue(), dequeue(), enqueue(4), dequeue(), dequeue().

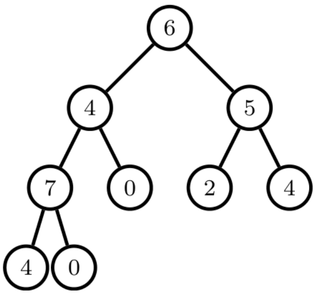
Q3. [Mod and div] Calculate the following:

1. 20 div 3 = ? 20 mod 3 = ?
2. 21 div 7 = ? 21 mod 7 = ?

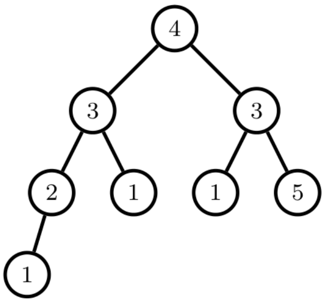
Q4. Decide which of the trees are heap trees.



(i)



(ii)

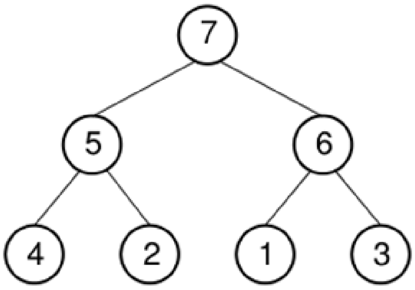


(iii)

Q5. Show step by step working of Insertion sort algorithm for the given array A

(5, 2, 4, 6, 1, 3)

Q6. Show the heap that results from deleting the maximum value from the max-heap of following Figure.



Q7. When writing code or text, it's important to have parentheses balanced and properly nested. For example:

- The string ((())())() has properly nested pairs of parentheses.
- The strings)()(and ()) do not have properly nested pairs of parentheses.

Identify the data structure you will use and write an algorithm that returns true if a string contains properly nested and balanced parentheses, and false if otherwise.

Q8. Let Q be a non-empty queue, and let S be an empty stack. Using only the stack and queue functions and a single element variable X, write an algorithm to reverse the order of the elements in Q.