Assume that you are given jobs j1, j2, . . . , j9, all with known running times t1, t2, . . . , t9, respectively. We have three processors p1,p2,p3. What is the best way to schedule these jobs in order to minimize the average completion time? We will assume non-preemptive scheduling: ie Once a job is started, it must run to completion. Identify the suitable data structure to solve the above problem and write the necessary insertion and deletion function.

Identify at least one algorithm for the following time complexities and describe about them in two lines and also sort them according to time complexity (fastest algorithm first).
O(1)
O(log n)
O(n^2)
O(n^3)
O(2^n)
O(n!)
O(n)
O(n log n)

Design a dynamic stack to do the following operations. Push the first element into the stack as it comes. When the second element has to be pushed, pop out the first and compare the two elements. Push the smaller one first and then the larger one. Follow the same procedure for the subsequent elements too.
Eg., 2,7,3,8,4,1,5

Push 2
Now 7 comes. Compare 7 and 2   So , push 2 and then 7
Next 3 comes. Compare 7 and 3. Push 3 and then 7
Next 8 comes. Compare 7 and 8. Push 7 and then 8
……….
Final output: 2,3,7,4,1,5,8