

## Exercise Sheet 8

Handout: Nov 7th — Deadline: Nov 14th 4pm

### Question 8.1 (0.25 marks)

Draw the following data structures after each of the following operations. Assume that the data structures are initially empty. You don't need to draw pointers for stacks and queues.

1. Consider a stack  $S$  and the operations  $\text{PUSH}(S, 7)$ ,  $\text{PUSH}(S, 4)$ ,  $\text{PUSH}(S, 5)$ ,  $\text{POP}(S)$ ,  $\text{PUSH}(S, 8)$ ,  $\text{POP}(S)$ ,  $\text{POP}(S)$ .
2. Consider a queue  $Q$  and the operations  $\text{ENQUEUE}(Q, 7)$ ,  $\text{ENQUEUE}(Q, 4)$ ,  $\text{ENQUEUE}(Q, 5)$ ,  $\text{DEQUEUE}(Q)$ ,  $\text{ENQUEUE}(Q, 8)$ ,  $\text{DEQUEUE}(Q)$ ,  $\text{DEQUEUE}(Q)$ .
3. Consider a singly-linked list  $L$  and the operations  $\text{LIST-PREPEND}(L, 7)$ ,  $\text{LIST-PREPEND}(L, 4)$ ,  $\text{LIST-PREPEND}(L, 5)$ ,  $\text{LIST-DELETE}(L, 4)$ ,  $\text{LIST-PREPEND}(L, 8)$ ,  $\text{LIST-DELETE}(L, 7)$ ,  $\text{LIST-DELETE}(L, 8)$ .

**Question 8.2** (0.5 marks) Explain how to implement two stacks  $S_1$  and  $S_2$  in one array  $A[1 : n]$  in such a way that neither stack overflows unless all the  $n$  elements of  $A$  are full. Present the pseudocodes for operations  $\text{PUSH}_{S_1}(A, x)$ ,  $\text{PUSH}_{S_2}(A, x)$ ,  $\text{POP}_{S_1}(A)$ , and  $\text{POP}_{S_2}(A)$ .

**Question 8.3** (0.25 marks) Rewrite  $\text{ENQUEUE}$  and  $\text{DEQUEUE}$  to detect underflow and overflow of a queue.

**Question 8.4** (0.5 marks) Show how to implement a Queue using 2 stacks  $S_1$  and  $S_2$ . Provide the pseudo-code of the operations  $\text{ENQUEUE}$  and  $\text{DEQUEUE}$ . You don't need to check for underflow and overflow. Analyse the runtime of the two operations.

**Question 8.5** (0.5 marks) Implement an algorithm that takes in input a mathematical formula with the following brackets  $()$ ,  $[]$ ,  $\{\}$ , and uses a stack to check whether the brackets are appropriately balanced or not.

**Question 8.6** (0.5 marks) Implement an Integer Calculator that takes a postfix expression in input using integers as operands and  $\{+, -, *\}$  as operators. The algorithm should use a stack.

**Question 8.7** (0.5 marks) Implement a scheduler for a shared printer that accepts requests for files to be printed and uses a queue to process the jobs in the order they have been requested as soon as the previous job is completed. The queue is implemented with an array  $A[1, 10]$  and the input contains either the job number 1, 2... or a signal 0 indicating that the previous print operation has completed.

Example input: 1 2 3 0 4 5 0 0 6 7 8 0 0 0 9 10 11 12 0 0 0 0 0 0 0