You are allowed to use copies of the slides during the exam.

1. You have two (unbounded) queues at our disposal, with the operations as specified in the abstract data type for queues.
   Use them to implement the operations push and pop of the abstract data type for (unbounded) stacks.
   Analyze the time complexity of push and pop in terms of $O$. (20 pts)

2. Apply heapsort to sort the following sequence of numbers.
   $$8 \ 17 \ 3 \ 28 \ 9 \ 12 \ 23 \ 5$$
   Show how the heap datastructure evolves during the computation. (20 pts)

3. Provide a pseudocode description of the mergesort algorithm.
   Use a recursive call for splitting the sequence of numbers into two halves and sorting those, if the sequence has size greater than one. Next call and specify a separate subroutine for merging two sorted sequences of numbers into one sorted sequence. (20 pts)

4. We consider a hash table of length 6, and the hash function $h(k) = k \mod 6$. Add (in the given order) the numbers
   $$1 \ 3 \ 2 \ 13 \ 9$$
   to an initially empty hash table, where collisions are resolved by:
   (a) chaining (6 pts)
   (b) open addressing with linear probing (6 pts)
   (c) open addressing with double hashing, using $h'(k) = 5 - (k \mod 5)$ (8 pts)

5. Apply Krusal’s algorithm to the graph below to compute the minimum spanning tree.

Show how the disjoint-set datastructure is maintained during the computation. (20 pts)