1. State a suitable loop invariant, and use it to prove that the following code returns the product of $x$ and $y$.

**Input:** nonnegative integers $x, y$

1. if $y = 0$ then return 0;
2. $z \leftarrow 0$;
3. while $y > 0$ do
   4. if even($y$) then
      5. $y \leftarrow y/2; x \leftarrow 2x$
   6. else
      7. $z \leftarrow z + x; y \leftarrow (y - 1)/2; x \leftarrow 2x$
   8. end
9. end
10. return $z$

What’s its time complexity?

For extra points: assume now that $x, y$ and $z$ are arbitrarily big integers, represented by stacks of binary digits — with the least significant bit on the top of the stack. Observe that Push($X, 0$) corresponds to the multiplication of the number represented by $X$ by 2. Similarly Pop corresponds to the division of $X$ by 2. Rewrite the pseudocode so it works also for these structures. What’s the time complexity now?

2. Let a sequence $a_1, a_2, \ldots, a_n$ be stored in a queue. Write a pseudocode of a linear-time algorithm that returns the longest increasing consecutive subsequence $a_i, a_{i+1}, a_{i+2}, \ldots, a_j$, again as a queue. Access the data only by using Enqueue and Dequeue.

You may use several queues, if needed.

3. **Towers of Hanoi.**

   There is a history about an Indian temple in Kashi Vishwanath which contains a large room with three time-worn posts in it surrounded by 64 golden disks. Brahmin priests, acting out the command of an ancient prophecy, have been moving these disks since that time, in accordance with the immutable rules of the Brahma:
   - Only one disk may be moved at a time.
   - Each move consists of taking the upper disk from one of the rods and sliding it onto another rod, on top of the other disks that may already be present on that rod.
   - No disk may be placed on top of a smaller disk.

   According to the legend, when the last move of the puzzle will be completed, the world will end.

   [From Wikipedia, the free encyclopedia]

Write pseudocode for a recursive procedure to move the the tower from one rod to another by using stacks.

You may compute the time complexity and determine how long would it take till the world end if each move would take only one second.

4. Write pseudocode for the recursive MERGESORT by using dynamic sets: either queues or stacks. Justify your choice of the data structure.

   - Assume that the input is given as a pointer to the data structure, s.t. repetitive use of DELETE (i.e. Pop or Dequeue depending on your choice) reveals the sequence.
   - Assume also that the number of elements to sort is a part of the input.