Decrease-by-constant-factor algorithms

- Read 4.4, pages 150-155
  - Study the **BinarySearch** algorithm in detail.
    * Explain the meaning of the variables l, r and m.
    * Explain the assignments to l and r inside the inner `if`.
    * How would we modify the algorithm, if the array was sorted in the reverse order (from largest to smallest)?
  - What is the best-case running time for the binary search algorithm? When does it occur?
  - What is the worst-case running time for the binary search algorithm? Develop it by building a recurrence relation.
  - Explain how the **Russian Peasant Multiplication** algorithm works.
    * Demonstrate its use if for $n = 45$ and $m = 126$.
    * The algorithm could start by possibly switching the roles of n and m. Does the choice of which of the two numbers is n and which is m affect the running time of the algorithm?
    * Write pseudocode for the RPM algorithm using a recursive approach. What about a non-recursive solution?
  - Read up on the Josephus problem.
    * Manually work out what would happen in the case where $n = 8$ and $n = 9$.
    * Make sure to understand the two recurrence relations that determine the relation between the person's position before and after a round of eliminatoins.
    * Exercise 4.4.15:
      · Directly compute $J(n)$ for each n from 1 to 15 by following the game rules.
      · Explain why $J(n) = 1$ for every power of 2.
      · Verify that the 1-bit cyclic shift of n does result in $J(n)$ for those cases you just computed.
      · Prove that the 1-bit cyclic shift operation obeys the same recurrence relations that $J(n)$ does, with the same start values when $n = 1, 2$.