Sections 5.3 et 5.4

1. Calculate the first few values of each of the following functions (e.g. $n = 1, 2, 3, \dots$).

   (a)
   $$f(n) = \begin{cases} 1 & \text{if } n = 1, \\ f(n-1) & \text{otherwise.} \end{cases}$$

   (b)
   $$f(n) = \begin{cases} 20 & \text{if } n = 1, \\ \frac{1}{2} f(n-1) & \text{if } f(n-1) \text{ is even,} \\ 3f(n-1) + 1 & \text{if } f(n-1) \text{ is odd.} \end{cases}$$

   (c)
   $$f(n) = \begin{cases} 1 & \text{if } n = 1, \\ n \cdot f(n-1) & \text{otherwise.} \end{cases}$$

   (d)
   $$f(n) = \begin{cases} 1 & \text{if } n = 1, \\ 11 \cdot f(n-1) & \text{otherwise.} \end{cases}$$

   (e)
   $$f(n) = \begin{cases} 1 & \text{if } n = 1, \\ 2 & \text{if } n = 2, \\ \frac{f(n-2)}{f(n-1)} & \text{otherwise.} \end{cases}$$

2. Find an equation for each of the functions in the previous question and prove that you answer is correct.

3. Define the following sets recursively. Show that you answer is correct.
   (a) The set of even positive integers.
   (b) The set of positive multiples of 5.
   (c) The set of powers of 3.
   (d) The set of perfect squares.

4. What is the set defined recursively by
   — $S \subseteq \mathbb{Z}^2$
   — $(0, 0) \in S$
   — If $(a, b) \in S$, then $(a, b + 1) \in S$.
   — If $(a, b) \in S$, then $(a + 1, b + 1) \in S$.

   Show that your answer is correct.

5. What is the set defined recursively by
   — $S \subseteq \mathbb{Z}^+ \times \mathbb{Z}^+$
   — $(0, 1) \in S$
   — $(1, 0) \in S$
   — If $(a, b) \in S$, then $(a + 2, b) \in S$.
   — If $(a, b) \in S$, then $(a, b + 2) \in S$.

   Show that your answer is correct.
6. What is the set defined recursively by
   — $8 \in S$
   — $6 \in S$
   — If $a, b \in S$, then $a + b \in S$
   — If $a, b \in S$, then $a - b \in S$

   Show that your answer is correct.
7. Write a recursive version of the pseudocode for a binary search.
   — Identify the base case.
   — How many sub-problems do we divide the input into ?
   — How many recursive calls do we make with these sub-problems ?
   — What is the merge step ?
8. Write a recursive version of the pseudocode for a sequential search.
   — Identify the base case.
   — How many sub-problems do we divide the input into ?
   — How many recursive calls do we make with these sub-problems ?
   — What is the merge step ?
9. Find a recursive algorithm that calculates the number of "1"s in a chain of binary characters.
   — Identify the base case.
   — How many sub-problems do we divide the input into ?
   — How many recursive calls do we make with these sub-problems ?
   — What is the merge step ?
10. Find a recursive algorithm that calculates the length of the longest sequence of consecutive "1"s in in a chain of binary characters.
   — Identify the base case.
   — How many sub-problems do we divide the input into ?
   — How many recursive calls do we make with these sub-problems ?

— What is the merge step ?

11. Find a recursive algorithm that calculates the length of the longest sequence of the form "$..., 0, 1, 0, 1, 0, 1, 0, 1, 0, ...$" in a chain of binary characters.
    — Identify the base case.
    — How many sub-problems do we divide the input into ?
    — How many recursive calls do we make with these sub-problems ?
    — What is the merge step ?
12. Find a recursive algorithm that calculates the smallest element in a table of numbers.
    — Identify the base case.
    — How many sub-problems do we divide the input into ?
    — How many recursive calls do we make with these sub-problems ?
    — What is the merge step ?
13. Find a recursive algorithm that calculates the second smallest element in a table of numbers.
    — Identify the base case.
    — How many sub-problems do we divide the input into ?
    — How many recursive calls do we make with these sub-problems ?
    — What is the merge step ?
14. Find a recursive algorithm that calculates the greatest common divisor of two numbers.
    — Identify the base case.
    — How many sub-problems do we divide the input into ?
    — How many recursive calls do we make with these sub-problems ?
    — What is the merge step ?
15. Find a recursive algorithm that calculates the smallest common multiple of two numbers.
    — Identify the base case.
    — How many sub-problems do we divide the input into ?
    — How many recursive calls do we make with these sub-problems ?
    — What is the merge step ?
16. Find a recursive algorithm that calculates the n^{th} Fibonacci number.
    — Identify the base case.
    — How many sub-problems do we divide the input into ?
    — How many recursive calls do we make with these sub-problems ?
    — What is the merge step ?

17. Find a recursive algorithm that calculates the sum of all the elements in a table of numbers.
    - Identify the base case.
    - How many sub-problems do we divide the input into ?
    - How many recursive calls do we make with these sub-problems ?
    - What is the merge step ?
18. Find a recursive algorithm that calculates the average of all the elements in a table of numbers.
    - Identify the base case.
    - How many sub-problems do we divide the input into ?
    - How many recursive calls do we make with these sub-problems ?
    - What is the merge step ?