X4: Algorithms and Complexity

Please submit your answers (as an *.rkt* file) via e-mail within a week (Subject should be: DCB X4). Keep in mind that the file must run without errors, and any procedures (names, arguments etc.) specified in a given task must be maintained.

Note: for each successfully completed lab-report you get a BONUS point that will be added on the final written examination points you will score.

Task 1

Fibonacci's model of rabbit expansion: One pair of adult rabbits creates a new pair of rabbits in the same time that it takes bunnies to grow into adults (i.e. one year). Thus the total number of rabbit pairs **T** at time *n* is $\mathbf{F_n + F_{n-1} + F_{n-2}}$**, where** $\mathbf{F_1 = 0, F_2 = 1}$. The intuition behind this is that the number of adult rabbits at time *n* is the number of rabbits (adults and babies) at time *n-1,* i.e. F_{n-1}, while the number of baby rabbits at time *n* is the number of adult rabbits at time n-1, which is F_{n-2}.

Propose a more realistic model of the rabbit life (and death) that limits the life span of rabbits by *k = 2.999* years. Then the corresponding sequence grows more slowly than the Fibonacci sequence.

a) Write the recurrence model of the realistic Fibonacci's model of rabbit expansion.
b) Write a procedure named *fibonacci-bounded* that will return the total number of rabbits for each generation up to n. You should implement both a recursive and an iterative procedure.

   Procedure
   ```
   fibonacci-bounded: number -> (list of numbers)
   ```

   Example:
   ```
   (fibonacci-bounded-rec 11)
   ;Value:
   (2 3 4 5 7 9 12 16)
   (fibonacci-bounded-it 11)
   ;Value:
   (2 3 4 5 7 9 12 16)
   ```
c) Provide the upper bound of the algorithm under these assumptions?

Task 2

Let x = n.

Is *log n = O(x)?*

Is *log n =Ω(x)?*

Is *log n = Θ(x)?*

If the answer is "no" to any of the questions, restate the question by changing *x* so that the answer is "yes".

Task 3

Write a procedure that returns every index combination from (0, 0, ..., 0) to (n_1, n_2, ..., n_d). What application do you see for this algorithm?

Procedure:

index: list -> (list of $(1+n_1)*(1+n_2)*...*(1+n_d)$ lists)

Example:

```
(index '(1 2 2)) ;Value: ((0 0 0) (0 0 1) (0 0 2) (0 1 0) (0 1 1) (0
1 2) (0 2 0) (0 2 1) (0 2 2) (1 0 0) (1 0 1) (1 0 2) (1 1 0) (1 1 1)
(1 1 2) (1 2 0) (1 2 1) (1 2 2))
```