Notes

- Recursive functions
    - Calls itself again to repeat the code
    - Always have another operation after the recursive call
        - E.g. (inc (p1 (dec a) b))
            - Where p1 is the recursive call to itself
            - Inc is the other operation (the output of the recursive call is the input for Inc)
            - Operations can be arithmetic, function call
    - Grows and shrinks
    - Has hidden information maintained by the interpreter and not contained in program variables
    - Control variable decides termination
    - “Solve a problem by breaking it into smaller pieces until it can be solved”
- Iterative functions
    - Repeats some part of the code
    - Never have another operation after the recursive call (only has one recursive call)
        - E.g. (p2 (dec a) (inc b))
            - Where p2 is the recursive call
    - Does not grow or shrink → makes code longer
    - Program description provides a complete description of the state of the process at any point
    - Conditional decides termination
    - “Repeating a task until it is done”
- Recursive procedure vs recursive process
    - Procedure
        - In the procedure definition (define), it calls itself
        - Contains at least one recursive call
        - Must have at least one base case
    - Process
        - A chain of deferred (postponed) operations
        - It has an operation to which a recursive call is an input. It cannot complete until that recursive call completes.
        - It expands and then contracts

        - It’s parameters do not keep track of the state of the process at all times
- Order of Growth
    - Provides a useful indication of how we may expect the behavior of the process to change as we change the size of the problem
        - Linear $\Theta(n)$: doubling the size will double the amount of resources used
        - Exponential $\Theta(n^2)$: each increment in problem size will multiply the resource utilization by a constant factor
- Invariant quantity
    - Use a new variable to keep track of the state
    - $a*b^n$ is unchanged from state to state
    - This helps with changing recursive processes to iterative processes
- Tail recursion
    - Recursive call is the last thing executed by the function
- Summary
    - Recursive: calling the same function
        - Deferred operations
        - Grows and shrinks
        - Must call itself
        - Conditional
        - Another operation after recursive call (can be multiple recursive calls)
    - Iterative: repetition of a block of code
        - Immediate operations
        - Makes code longer
        - Calls a part of the code
        - Control/state variables
        - Only one recursive call (usually no operation after)
    - A function can be a recursive procedure but not a recursive process