Project #1 – Warm-up with Racket Lists

Learning Objectives

- Write functional code in Racket
- Understand and use Scheme-style structures in Racket

Overview

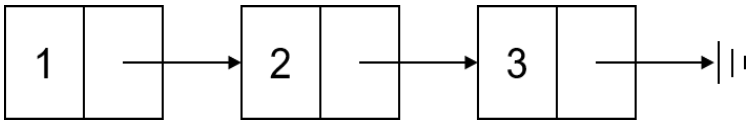
This short project provides an introduction to Racket forms, lists, functions, and the fundamentals of functional programming

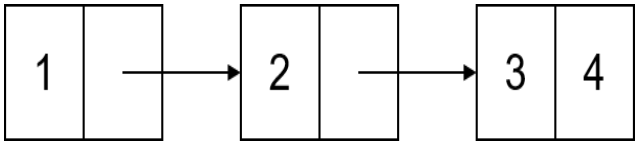
Tasks

Part 1: (10 points): Using only the `(cons )` function, write a Racket form that returns each of the pair constructs below:

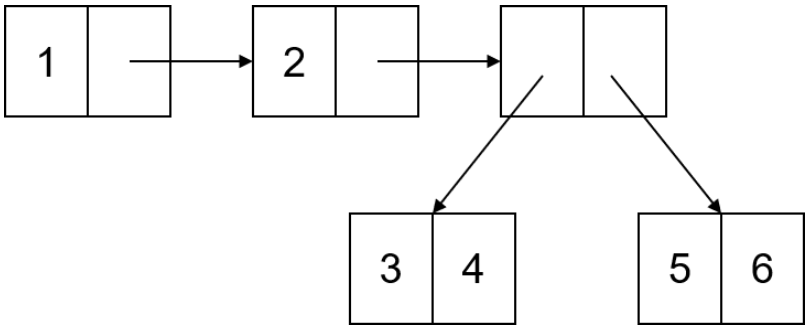
1. **`(1 2 3 4)`**
2. **`(1 (2 . 3) 4)`**
3. **`(1 2 (3 (4 . 5)))`**
4. **`(1 2 (3 4 5))`**
5. **`((1 . 2) (3 . 4) . 5)`**

Part 2: (10 points): Using only the `(cons )` function, write a Racket form that returns the pair represented by each of the following box diagrams:

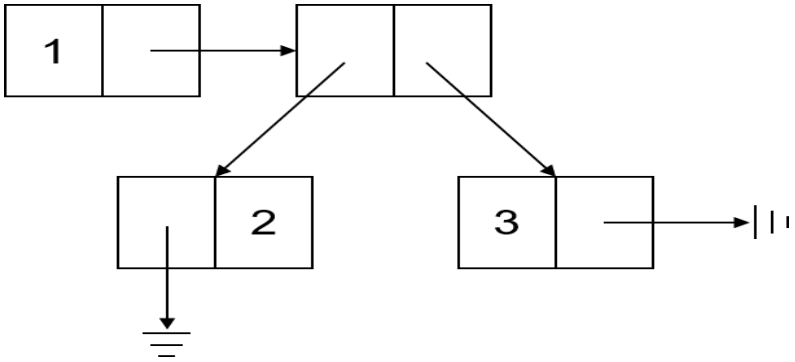
6. 

7. 

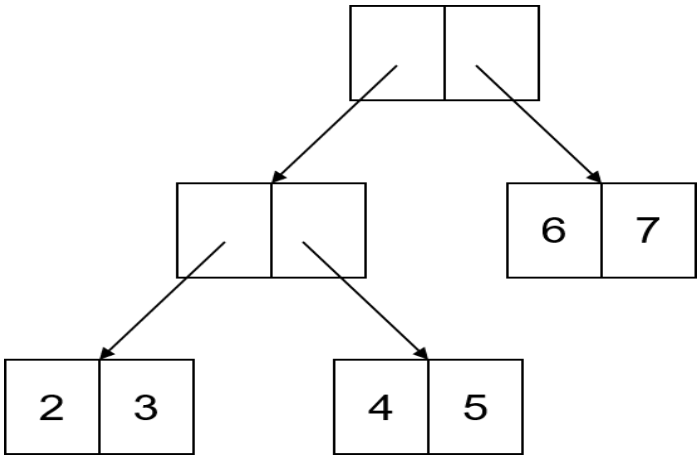
8.



9.



10.



Part 3: (10 points): write Racket functions:

- Write a function (**avg3**) that expects three numerical arguments and returns the average

- Write a function (**third-item**) that expects a list of arguments (any length > 3) and returns the third item in the list

Turn in and Grading

- Turn in a single Racket source file named **Project1.rkt** via the Pilot Dropbox. This file should contain precisely 10 Racket forms – one for each problem above (parts1, 2) and two functions.
- The project is worth 30 points. You will receive 2 points for each correct (cons) form and 5 point for each correct function.
- **Write your name (commented by ;) on the first line Immediately following the line #lang racket.**