

Assignment 7: Clojure

Assigned: Saturday, 5 March 2011

Due: 11:00 p.m., Saturday, 12 March 2011

Summary: In this assignment, you will explore various aspects of Clojure.

Purposes: To give you more experience with Clojure. To give you the opportunity to think about the implementation of data types.

Expected Time: Three to four hours.

Collaboration: I would prefer that you work on this assignment in groups of two or three. You may also work on this assignment alone or in groups of up to size four. You may consult with anyone you wish on this assignment (including other classmates), provided you cite your consultants.

Submitting: Upload a tarball to Pioneerweb.

Warning: So that this assignment is a learning experience for everyone, I may spend class time publicly critiquing your work.

Assignment

a. Implement something like Clojure vectors in Java. Need to support

- Get: Constant time
- First: Constant time
- Rest: Constant time
- Cons: Creates another vector
- Constructor takes an array as a parameter

b. Lazy Fibonacci, the right way.

```
(def fib
  (cons 1 (cons 1
    (map + fib (rest fib))))))
```

c. Sieve of Eratosthenes

Implement it

d. Let equivalent (macro)

In Scheme,

```
(let ((v1 e1) (v2 e2)) body)
```

is shorthand for

```
((lambda (v1 v2) body) e1 e2)
```

Implement this macro in Clojure.

e. For loops

```
(for var lb ub body)
```

f. For-each loops

```
(for-each var lst body)
```

Important Evaluation Criteria

I will look at the standard “C” criteria: Is your code **correct**? Is your code **concise**? (This second attribute is particularly appropriate in Ruby.) Is your code appropriately **commented**? Have you used any particularly **clever** techniques. How **comely** is your code?

Submitting Your Homework

Create a directory for this assignment of the form `hw07.names`. Within that directory, create subdirectories for each part of the assignment (e.g., `part0`, `part1`). Each subdirectory should contain:

- The code that you have written for that part of the assignment.
- A typescript or typescripts in which you demonstrate the utility of the code you have written.
- (Optional) A README file that describes any particularly interesting aspects of your code, or other things that you want me to know.
- (Optional) Unit tests.

Make a gzipped tarball of the assignment directory. (If you don’t know how to do that, let me know.) Upload it via the Assignment 7 link on Pioneerweb.