STREAMS, ITERATORS, GENERATORS

COMPUTER SCIENCE MENTORS 61A

April 4 to April 9, 2016

1 Streams

1. Whats the advantage of using a stream over a linked list?

2. Whats the maximum size of a stream?

3. Whats stored in first and rest? What are their types?

4. When is the next element actually calculated?

2 What Would Scheme Print?

5. For each of the following lines of code, write what scheme would output.

```
scm> (define x 1)

scm> (if 2 3 4)

scm> (delay (+ x 1))

scm> (define (foo x) (+ x 10))

scm> (define bar (cons-stream (foo 1) (cons-stream (foo 2)
   bar)))

scm> (car bar)


scm> (cdr bar)

scm> (define (foo x) (+ x 1))

scm> (cdr-stream bar)

scm> (define (foo x) (+ x 5))

scm> (car bar)

scm> (cdr-stream bar)

scm> (define (foo x) (+ x 5))

scm> (car bar)

scm> (cdr-stream bar)
```

3 Code Writing for Streams

I

4 Iterators

)

s

10. Is this an iterator or an iterable or both?
11. (Optional) Make `Accumulator` work if it takes in any iterable, not just a list

5 Generators

)

14. (Optional) Define `tree`$_s equenceageneratorthatiteratesthroughatreebyfirstyieldingtherootvalueand$

`b` Tree(3, [Tree(4)])]) ¿¿¿ print list(tree$_s equence(tree))[1, 2, 5, 3, 4]$

Computer Science Mentors CS61A Spring 2016: Jerry Chen and Tiffany Perumpail, with
Albert Chen, Amanda Bui, Anthony Dela Paz, Anders Lewis, Ching Fang, Colin Schoen, Cynthia Shen, Kevin Lin,
Kimberly Bourque, Kunal Munshani, Meha Bakshi, Regina Xu, and Ryan Chang