CS 135 — L08: List Abbreviations & Big-O

Luke Lu • 2025-12-17

Website: student.cs.uwaterloo.ca/~cs135

List Abbreviations

(list 1 2 3) is shorthand for (cons 1 (cons 2 (cons 3 empty))).

Counting Steps → Big-O

Measure time by the **number of substitutions**. For linear list recursion, steps ∝ list length n.

Linear example:

```
(define (len lst)
  (cond [(empty? lst) 0]
        [else (add1 (len (rest lst)))]))
```

> Built-ins like length, append, and reverse should be treated as **linear** in the size of their input lists for complexity reasoning.

Spotting Quadratic Work

```
(define (rev-bad xs)              ; quadratic: uses append each step
  (cond [(empty? xs) empty]
        [else (append (rev-bad (rest xs)) (list (first xs)))]))
```

Exercises

1. Classify the complexity of map, filter, increasing?, insert.
2. Rewrite rev-bad using an accumulator to get linear time.