

Clicker Questions
for
February 13

```
(define reverse-a (lambda (lat acc)
  (cond
    [(null? lat) acc]
    [else (reverse-a (cdr lat) (cons (car lat) acc))])))
```

How many primitive ops (car cdr or cons) does it take this to reverse a list of length n ?

- A. 2
- B. $O(1)$
- C. $O(n)$
- D. $O(n^2)$
- E. $O(n^3)$

Answer C: $O(n)$

```
(define reverse (lambda (lat)
  (cond
    [(null? lat) acc]
    [else (append (reverse (cdr lat)) (list (car lat)))])))
```

How many primitive ops (car cdr or cons) does it take this to reverse a list of length n ?

- A. 2
- B. $O(1)$
- C. $O(n)$
- D. $O(n^2)$
- E. $O(n^3)$

Answer: It depends on how append is implemented. If it is just

```
(define append (lambda (L1 L2)
  (cond
    [(null? L1) L2]
    [else (cons (car L1) (append (cdr L1) L2))]))
```

then the answer is D: $O(n^2)$.

If the lists are doubly-linked so you can append L1 to L2 by making the tail of L1 point to L2 then append might be constant-time, so reverse would still be $O(n)$.