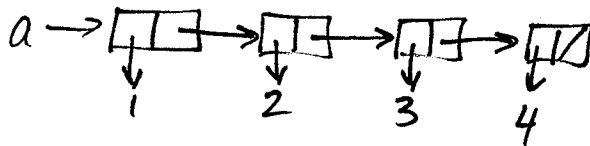


Quiz 1 Solutions

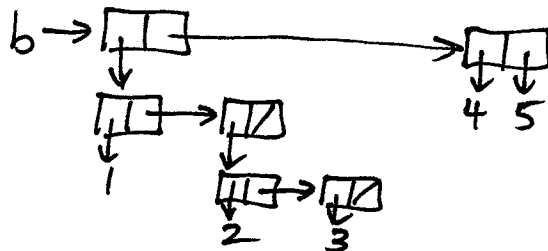
Problem 1

6
0
9
5
5

Problem 2



`(car (cdr (car a)))` or `(caddr a)`



`(car (cdr (car (cdr (car b)))))` or `(car (cdadar a))`

Problem 3

a. 7
b. 5

c. The code given in this problem is more efficient with applicative order than with normal order.

d.

```
(define (foo a b)
  (if #t
      a
      b))
```

In normal order, the item passed to b is never evaluated, but it would be with applicative order.

Problem 4

$T(n) = \Theta(n)$

$S(n) = \Theta(n)$

n is dependent upon the length of lst
recursive process

```
(define (replace-all2 x y lst)
  (define (replace-iter old-list new-list)
    (cond ((null? old-list) new-list)
          ((eqv? x (car old-list))
           (replace-iter (cdr old-list)
                         (append new-list (list y))))
          (else
           (replace-iter (cdr old-list)
                         (append new-list
                                  (list (car old-list)))))))
  (replace-iter lst '()))
```

```
(define (replace-all x y lst)
  (tree-accumulate lst
                    (lambda (z) (if (equal? z x) y z))
                    cons
                    nil))
```

Problem 5

```
(define (spiral painter n)
  (if (= n 0)
      painter
      (beside (below (rotate180 (spiral painter (- n 1))))
              (rotate90 painter)
              painter)))
```

Problem 6

```
(define (car p) (cadr p))
(define (cdr p) (caddr p))
```

```
(define (car p) (p 'car))
(define (cdr p) (p 'cdr))
```