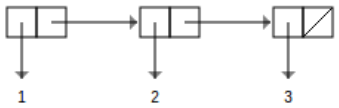
Quiz 7

Brian Hou

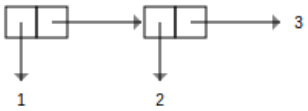
October 29, 2015

1. Draw box-and-pointer diagrams for the following.

```
scm> '(1 . (2 . (3)))
```



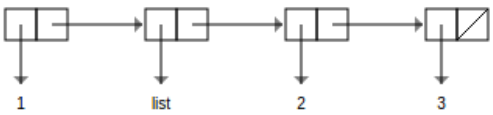

```
scm> '(1 2 . 3)
```



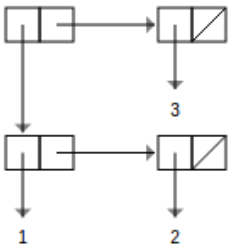

```
scm> '(1 . 2 . 3)
```

Error: you can only have one element after a dot. This is actually an impossible structure to make.

```
scm> (cons 1 '(list 2 3))
```




```
scm> (list (append '(1) '(2) nil) 3)
```



2. Write a function `take` that takes in a list `s` and a positive number `n`, and returns a list `t` such that `(car t)` is the first `n` elements of `s` and `(cdr t)` is the remaining elements of `s`. If `n` is greater than the length of `s`, `(car t)` should be `s` and `(cdr t)` should be `nil`.

```
(define (take s n)

  (cond ((= n 0) (cons nil s))
        ((null? s) (cons s nil))
        (else (let ((rec (take (cdr s) (- n 1))))
                (cons (cons (car s) (car rec)) (cdr rec))))))
```

```
scm> (define a (take '(1 2 3) 2))
scm> (car a)
(1 2)
scm> (cdr a)
(3)
scm> (define b (take '(1 2 3) 4)) ; n > (length s)
scm> (car b)
(1 2 3)
scm> (cdr b)
()
```