practice problem: memoization

Memoization is a clever idea that allows us to save on computation. It works by keeping track of evaluation of a procedure on a specific argument, and simply returns the remembered value if the procedure has already been run on that argument. Draw the environment diagram for the following procedure definition and application.

```
(define (memoize proc)
  (let ((cache '()))
    (lambda (arg)
      (if (assoc arg cache)
       (cadr (assoc arg cache))
       (let ((answer (proc arg)))
         (set! cache (cons (list arg answer) cache))
         answer)))))
(define my-sq (memoize (lambda (x) (* x  x))))
(my-sq 5)
```

memoization

Memoization is a clever idea that allows us to save on computation. It works by keeping track of evaluation of a procedure on a specific argument, and simply returns the remembered value if the procedure has already been run on that argument.

```
(define (memoize proc)
  (let ((cache '()))
    (lambda (arg)
      (if (assoc arg cache)
          (cadr (assoc arg cache))
          (let ((answer (proc arg)))
            (set! cache (cons (list arg answer) cache))
            answer)))))
(define my-sq (memoize (lambda (x) (* x x))))
(my-sq 5)
```