

Homework 4 - Mutations

本次作业需要在 [Online Judge](#) 上提交（校内网），校外可以使用[学校VPN](#)。服务器由于断电、重启等原因也可能短暂地不可用，请同学们在本地留存好自己的代码。

默认每题时限为 180 秒，内存限制 1024 MB，栈大小同内存限制，正确实现的代码可以以远低于该时空限制的要求下通过所有测试用例。提交答案需要完整地赋值黏贴对应题目的源码，不包括测试代码。每题满分均为 100，你的得分为 $\left\lfloor \frac{\#passed}{\#total} \right\rfloor$ ，其中 $\#passed$ 为通过的测试用例数量， $\#total$ 为该题的测试用例数量，以最后一次提交的得分为准。作业截止时间以 OJ 为准。

遇到问题可以联系助教 23110240130@m.fudan.edu.cn。

Note

除了函数式原教旨主义语言（如 Haskell），许多函数式语言都支持 Mutation（如 Racket、OCaml）。Mutation 虽然会带来副作用，但在一些场景下会很好用。

本次的作业中你需要：

1. 实现通用的表格 `table` 用于存储键值对（对 Mutation 的理解）
2. 利用 `table` 实现纯函数的记忆化（对 Environment Model 的理解）

不同于 R⁵RS，当前版本的 Racket 把不可变列表和可变列表区别开来了，与可变列表相关操作的名字发生了变化，与课上介绍的函数名略有不同，比如：`list` 变为 `mlist`；`cons` 变为 `mcons`；`pair?` 变为 `mpair?`；`car` 变为 `mcar`；`set-cdr!` 变为 `set-mcdr!` 等等。

想要深入了解可阅读：

1. [Getting rid of set-car! and set-cdr!](#)
2. [Mutable Pairs and Lists](#)

A memo

A.1 table

你需要基于结构体 `table` 实现一个表格用于存储键值对，为此需要实现以下操作：

1. `table-put!` 用于添加键值对；
2. `table-has-key?` 用于查询某个键是否存在；
3. `table-get` 用于获取某个键对应的值。

以上所有操作的时间复杂度均不做过高要求，只要是 $O(n)$ 的即可， n 为操作时表中元素个数。过高的复杂度会影响 `memoize` 的效率。

提示 你可以使用 `(set-table-data! t new-data)` 来更新 `t` 的 `data` 域。

A.2 memoize

考虑以下斐波那契数列的实现：

```
1 ; fib :: Int -> Int
2 ; Calculate the n-th term of the Fibonacci sequence (A000045) mod 19260817
```

```

3 (define (fib n)
4   (if (< n 2)
5       (remainder n mod)
6       (remainder (+ (fib (- n 1)) (fib (- n 2))) mod)))

```

不难注意到时间复杂度是 $\Omega(\text{Fib}_n)$ 的，即使是 `(fib 100)` 也难以计算，其原因在于相同的调用会被重复计算数次。

因为函数 `fib` 是纯函数，若参数 n 不变化，其函数值也不会变化，那么可以利用 `table` 把计算过的函数值缓存起来，避免重复计算。在 Python 里你可以通过 `lru_cache` 装饰器来轻松实现。

```

1 from functools import lru_cache
2
3 @lru_cache(maxsize=None)
4 def fibonacci(n):
5     if n < 2:
6         return n
7     return fibonacci(n - 1) + fibonacci(n - 2)

```

现在你需要通过 `table` 来实现 Racket 版本的把纯函数记忆化的高阶函数 `memoize`。

可以按以下方式把求 [斐波那契数列](#) 和 [纳拉亚纳奶牛数列](#) 转化为记忆化的函数，可阅读 `memo-test.rkt` 了解更多：

```

1 (set! fib (memoize fib))
2 (set! narayana (memoize narayana))

```

此时 `(fib 100)` 就能很快得出结果。

思考 把 `(set! fib (memoize fib))` 改成 `(define fib-memo (memoize fib))` 还对吗？

提示 结合 Environment Model (Lecture 4 Slides P40 ~ P68) 理解。

Reference

1. D. Friedman, M. Felleisen, *The little Schemer (4th ed.)*. MIT Press, 1996. [\[pdf\]](#)
2. H. Abelson, G. Sussman, *Structure and Interpretation of Computer Programs*. MIT Press, 1996. [\[pdf\]](#)