Scheme language

Introduction

Scheme is a programming language that supports multiple paradigms, including functional programming and imperative programming, and is one of the two main dialects of Lisp.

Implementation of scheme

There is three main scheme implementation Chez scheme, Racket, Chicken. Among these, racket provides a GUI and will be easy to use.

Test

They both provide REPL(Read–eval–print loop) where you can test each function. For example, in "bst.scm", we can insert-list function and racket will return the result.

```
racket@> (insert-list nil '(4 2 1 7 0 8))
'(4 (2 (1 (0 () ()) ()) ()) (7 () (8 () ())))
racket@> (define s '(4 (2 (1 (0 () ()) ()) ()) (7 () (8 () ()))))
racket@> (delete s 0)
'(4 (2 (1 () ()) ()) (7 () (8 () ())))
```

Our program

Our splay.exe, bst.exe, avl.exe will read "xxx-test.txt" and output the result. 0 stands for insert, 1 for delete and 2 for search.