

Отчет по лабораторной работе № 3 по курсу «Функциональное программирование»

Студент группы 8О-308 МАИ *Балес Александр*, №3 по списку
Контакты: `aleks_bales@mail.ru`
Работа выполнена: 01.05.2016

Преподаватель: Иванов Дмитрий Анатольевич, доц. каф. 806
Отчет сдан:
Итоговая оценка:
Подпись преподавателя:

1. Тема работы

Последовательности, массивы и управляющие конструкции Коммон Лисп.

2. Цель работы

Научиться конструировать матрицы, последовательности, выполнять различные операции над матрицами и последовательностями.

3. Задание(вариант 3.41)

Запрограммировать на языке Коммон Лисп функцию, принимающую в качестве аргумента двумерный массив, представляющий действительную квадратную матрицу \mathbb{A} порядка n .

Функция должна возвращать произведение матриц $\mathbb{A} \times \mathbb{B}$, где элементы матрицы \mathbb{B} вычисляются по формуле:

$$b_{ij} = \begin{cases} \frac{1}{(i+j-1)} & , \text{ если } i < j \\ 0 & , \text{ если } i = j, \text{ где } i, j = \overline{1, n} \\ \frac{-1}{(i+j-1)} & , \text{ если } i > j \end{cases}$$

4. Оборудование студента

Процессор Intel Core i5-3210 4 @ 2.5GHz, память: 8192Mb, разрядность системы: 64.

5. Программное обеспечение

ОС Ubuntu 14.04, среда GNU Common Lisp 2.6.10

6. Идея, метод, алгоритм

При помощи ф-ий `setf`, `loop`, `array-dimension`, `aref` инициализируем матрицу \mathbb{B} , а затем запускаем тривиальный алгоритм перемножения матриц, работающий за $O(N^3)$.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

```
(defun mult-matrix (A B)
  (let ((matrixRes (make-array (list (array-dimension A 0)
    (array-dimension B 1))
    :initial-element 0.0
    :element-type 'single-float)))
    (loop with sz1 = (array-dimension A 0)
      for i upfrom 0 below sz1 do
        (loop with sz2 = (array-dimension A 1)
          for j upfrom 0 below sz2 do
            (let ((temp 0))
              (loop with sz3 = (array-dimension B 0)
                for k upfrom 0 below sz3 do
                  (setq temp (+ temp (* (aref A i k) (aref B k
j))))))
              (setf (aref matrixRes i j) temp))))
    matrixRes))

(defun task (matrixA matrixB)
  (loop with n = (array-dimension matrixA 0)
    for i from 1 to n do
      (loop with szRow = (array-dimension matrixA 1)
        for j from 1 to szRow do
          (cond ((= i j) (setf (aref matrixB (- i 1) (- j 1))
0.0))
            ((< i j) (setf (aref matrixB (- i 1) (- j 1)) (/
1.0 (+ i (- j 1)))))
            (t (setf (aref matrixB (- i 1) (- j 1)) (/ -1.0 (+
i (- j 1)))))))
    (mult-matrix matrixA matrixB))

(defun start (matrixA)
```

```

(let ((matrixB (make-array (list (array-dimension matrixA 0)
    (array-dimension matrixA 1)))))
    (task matrixA matrixB)))

(defun init-matrix (matr cntRow cntCol)
  (let ((val 0.0))
    (loop for i upfrom 0 below cntRow do
      (loop for j upfrom 0 below cntCol do
        (setf (aref matr i j) (setq val (1+ val))))))
    (values))

(defun main (n)
  (let ((matrixA (make-array (list n n)
    :element-type 'single-float)))
    (init-matrix matrixA n n)
    (PrintMatrix (start matrixA))))

(defun GetListFromMatrix (A row curPos len)
  (if (>= curPos len) ()
    (cons (aref A row curPos) (GetListFromMatrix A row (1+
    curPos) len))))

(defun PrintMatrix (A)
  (loop with n = (array-dimension A 0)
    for i upfrom 0 below n do
      (pprint (GetListFromMatrix A i 0 n))))

```

8.1. Результаты

```
bai@bai-HP-ProBook-4540s:~/src/GitHub/FunctionalProgramming/labs/lab3$ gcl -load lr3.lsp
GCL (GNU Common Lisp) 2.6.10 CLtL1 Apr 2 2014 14:22:53
Source License: LGPL(gcl,gmp), GPL(unexec,bfd,xgcl)
Binary License: GPL due to GPL'ed components: (XGCL READLINE UNEXEC)
Modifications of this banner must retain notice of a compatible license
Dedicated to the memory of W. Schelter

Use (help) to get some basic information on how to use GCL.
Temporary directory for compiler files set to /tmp/

>(main 1)

(0.0)
NIL

>(main 2)

(-1.0 0.5)
(-2.0 1.5)
NIL

>(main 3)

(-2.0 -0.25 0.8333333333333333)
(-4.5 0.5 2.583333333333333)
(-7.0 1.25 4.333333333333333)
NIL
```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
11.05.2016	BugFix	Изменил ограничение в одном из циклов при перемножении матриц (самый вложенный), не смотря на то, что это никак не влияло на результат, т.к. матрицы квадратные.	Исправления были добавлены для унификации функции произведения матриц.
11.05.2016	Refactoring	Заменял (- 1.0) на -1.0	
11.05.2016	Refactoring	Вынес в глобальную область ф-ию GetListFromMatrix	
11.05.2016	Refactoring	Убрал вшитую в код размерность матрицы, теперь можно подавать на вход программе (main n)	

10. Замечания, выводы

Мною была также написана ф-ия для «красивого» вывода матрицы `PrintMatrix` (на мой взгляд). В результатах работы программы матрица $\mathbb{A}_{3 \times 3}$.

$$\mathbb{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \mathbb{B} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{3} \\ -\frac{1}{2} & 0 & \frac{1}{4} \\ -\frac{1}{3} & -\frac{1}{4} & 0 \end{bmatrix}$$

Для демонстрации корректности результата, я прикрепляю скриншот с сайта, на котором реализован алгоритм произведения матриц:

Решение:

$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0.5 & 0.333 \\ -0.5 & 0 & 0.25 \\ -0.333 & -0.25 & 0 \end{pmatrix} = \begin{pmatrix} -1.999 & -0.25 & 0.833 \\ -4.498 & 0.5 & 2.582 \\ -6.997 & 1.25 & 4.331 \end{pmatrix}$$