

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

Студент группы 8О-307 МАИ *Бронников Максим*, №4 по списку

Контакты: `max120199@gmail.com`

Работа выполнена: 01.05.2020

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 создавать векторы и массивы для представления матриц, освоить общие функции работы с последовательностями, инструкции цикла и нелокального выхода.

3. Задание (вариант №14)

Запрограммировать на языке Коммон Лисп функцию, принимающую два аргумента:

- A - двумерный массив, представляющий действительную матрицу произвольного размера
- r - действительное число.

Функция должна найти наименьший элемент матрицы и вернуть новую матрицу того же размера, получающуюся из данной заменой всех вхождений наименьшего элемента на r .

Исходный массив должен оставаться неизменным.

4. Оборудование ПЭВМ студента

Процессор Intel© Celeron© CPU @ 2.16GHz x 2, память: 4096Mb, разрядность системы: 64.

5. Программное обеспечение ЭВМ студента

OS Linux Mint 19.3 Cinnamon, среда SLIME 2.24 с реализацией языка SBCL 1.4.5.debian

6. Идея, метод, алгоритм

Функции `first_size` и `second_size` вызывают метод `array-dimensions` массива, который возвращает список - размеры массива по его измерениям, и возвращают первый и второй элемент этого списка соответственно.

Функция `search_min` проходит по массиву и ищет минимальный элемент. При этом если массив нулевого размера, неважно какое значение возвращать. Начальное значение инициализируется первым элементом массива, после чего идет проход, при котором элемент массива сравнивается с текущим минимумом и если значение меньше, то минимум заменяется этим значением.

Функция `fill_matrix` берет аргументами значение r , минимум исходного массива A , массив A и тот, который нужно заполнить в соответствии с требованием задания, проходит по массивам и помещает в заполняемый массив значение массива A , если оно больше минимума, иначе помещает значение A .

Функция `change_min_matrix_to_num` - главная функция. Она принимает аргументы A и r , создает массив такого же размера, как и A , после чего находит минимум и вызывает функцию `fill_matrix` для заполнения возвращаемого массива.

7. Сценарий выполнения работы

8. Распечатка программы и её результаты

8.1. Исходный код

```
;;; array-dimensions returns list (first second)

;;; func returns first size of matrix
(defun first_size (A)
  (car (array-dimensions A)))

;;; func return second size of matrix
(defun second_size (A)
  (car (cdr (array-dimensions A))))

;;; if matrix size with 0 => return any value(0 for example)
;;; else search min, start value - A[0][0]
(defun search_min (A)
  (if (or (< (first_size A) 1) (< (second_size A) 1))
```

```

0
(let ((ans (aref A 0 0)))
  (dotimes (i (first_size A))
    (dotimes (j (second_size A))
      (if (< (aref A i j) ans)
          (setf ans (aref A i j))))))
  ans)))

;;; fill ans by requirement, m - minimum in A, r - new value
(defun fill_matrix (ans A m r)
  (dotimes (i (first_size A))
    (dotimes (j (second_size A))
      (if (> (aref A i j) m)
          (setf (aref ans i j) (aref A i j))
          (setf (aref ans i j) r))))
  ans)

;;; create matrix, search min, fill this matrix and return
(defun change_min_matrix_to_num (A r)
  (let ((ans (make-array (array-dimensions A))))
    (fill_matrix ans A (search_min A) r)))

```

8.2. Результаты работы

```

(base) max@max-Lenovo-B50-30:~/FuncProg/lab3$ ls
main.lisp
(base) max@max-Lenovo-B50-30:~/FuncProg/lab3$ sbcl
This is SBCL 1.4.5.debian, an implementation of ANSI Common Lisp.
More information about SBCL is available at
<http://www.sbcl.org/>.

```

SBCL is free software, provided as is, with absolutely no warranty.

It is mostly in the public domain; some portions are provided under

BSD-style licenses. See the CREDITS and COPYING files in the distribution for more information.

```
* (load "main.lisp")
```

T

```
* (defvar test_array (make-array '(3 3)
:initial-contents '((0 1 2) (3 4 5) (6 7 8))))
```

```

TEST_ARRAY
* (change_min_matrix_to_num test_array 10)

#2A((10 1 2) (3 4 5) (6 7 8))
* test_array

#2A((0 1 2) (3 4 5) (6 7 8))
* (setf (aref test_array 1 2) -6)

-6
* (setf (aref test_array 0 1) -6)

-6
* test_array

#2A((0 -6 2) (3 4 -6) (6 7 8))
* (change_min_matrix_to_num test_array 3.543)

#2A((0 3.543 2) (3 4 3.543) (6 7 8))
* (setf test_array (make-array '(0 0)))

#2A()
* (change_min_matrix_to_num test_array -3)

#2A()
* (setf test_array (make-array '(2 5)
:initial-contents '((0 1 2 -6 -13.32) (1.21 -5.5442 -13.32 54
-13.32)))
)

#2A((0 1 2 -6 -13.32) (1.21 -5.5442 -13.32 54 -13.32))
* (change_min_matrix_to_num test_array -25)

#2A((0 1 2 -6 -25) (1.21 -5.5442 -25 54 -25))
* test_array

#2A((0 1 2 -6 -13.32) (1.21 -5.5442 -13.32 54 -13.32))
*
(base) max@max-Lenovo-B50-30:~/FuncProg/lab3$ ls
main.lisp
(base) max@max-Lenovo-B50-30:~/FuncProg/lab3$ exit

```

9. Дневник отладки

Дата	Событие	Действие по исправлению	Примечание
------	---------	-------------------------	------------

10. Замечания автора по существу работы

Информация о том, что массив, передаваемый в качестве аргумента, имеет размер $N \times M$, где $N > 0$, $M > 0$ упростила бы написание кода и сделало бы решение более элегантным.

11. Выводы

В процессе выполнения работы я познакомился с новым для себя типом данных в *Common Lisp* - массивом, узнал как объявлять и заполнять значения переменных и даже написал простейшую программу с использованием стандартных функций работы с массивами.

Вспомогательные функции `search_min` и `fill_matrix` работают за квадратичное время, поскольку делают один проход по массиву с проверкой и заполнением за константное время. Также есть функции `first_size` и `second_size`, которые работают за время, не превышающее $O(n^2)$. Значит основная функция осуществляет свою работу за время $O(n^2)$, поскольку последовательно вызывает в себе функции с квадратичным временем работы.

Наибольшую трудность при выполнении работы мне доставило осознание того, как работает функция `let`, которая была необходима для нахождения минимального значения массива.

Эта задача заставила меня поломать голову, поскольку мне было непривычно перестроиться на ее выполнение после выполнения лабораторной работы по другому предмету на языке *Python*, однако я считаю полезной такую разминку мозга и надеюсь на не менее трудные задания в дальнейшем.