Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Лабораторная работа №4
по дисциплине: «Теория Алгоритмов»
Тема: «Бинарные деревья и графы»

Выполнила ученица 223 группы
Никончук А.П.

Проверил
Сорокин Д.С. _____

Санкт-Петербург,
2014 г.

Задание 1: написать функцию, которая строит дерево выражений

```
def build_exp_tree(exp_string):
    ''' Построить дерево выражений по строке exp_string '''

    #если строка это узел, то вернётся конечный вариант
    if st >= '0' and st <= '9':
        return [int(st), [], []]

    br = 0
    idx = 1

    # Если все скобки закрыты, и текущий символ знак - найти знак
    while True:
        if br == 0 and st[idx] in signs:
            break

        if st[idx] == '(': br += 1
        elif st[idx] == ')': br -= 1
        idx += 1

    tree = binary_tree(st[idx])
    insert_left(tree, build_exp_tree(st[1:idx]))
    insert_right(tree, build_exp_tree(st[idx+1:-1]))

    return tree
```

```
('String', '((((3+1)*3)/((9-5)+2))-((3*(7-4))+6))')
('Tree: ', ['-', ['/', ['*', ['+', [3, [], []], [1, [], []]], [3, [], []]], ['+'
, ['-', [9, [], []], [5, [], []]], [2, [], []]]], ['+', ['*', [3, [], []], ['-',
 [7, [], []], [4, [], []]]], [6, [], []]])
('Exp Result: ', -13.0)
```

Задание 2.1: написать функцию, которая выводит лабиринт на экран

```
def print_maze(maze):
    ''' Вывод лабиринта на экран '''
    for i in range (len(maze)):
        for j in range(len(maze[0])):
            print(maze[i][j]),
        print
    print
```

Задание 2.2: написать функцию, которая создаёт граф из лабиринта

```
def maze2graph(maze):
    def cell_to_graph(cell):
        idxs = [(0, -1), (-1, 0), (1, 0), (0, 1)]
        for idx in idxs:
            if cell[0]+idx[0] in range(len(maze)) and cell[1]+idx[1] in range(len(maze[0])) and \
            maze[cell[0]+idx[0]][cell[1]+idx[1]] in [' ', 'G'] and \
            (cell[0]+idx[0], cell[1]+idx[1]) not in G[cell]:
                        add_edge(G, ( cell, (cell[0]+idx[0], cell[1]+idx[1])))
                        cell_to_graph( (cell[0]+idx[0], cell[1]+idx[1]))


    G = make_graph()
    add_vertex(G, (0, 0) )
    cell_to_graph((0, 0))
    return G


print(maze2graph(maz))
```

```
Администратор@TATYANNA ~/cs101myworks/lab_4 (master)
$ python lab_4_2_1.py
{(1, 2): [(1, 1), (0, 2), (2, 2), (1, 3)], (2, 0): [(1, 0), (3, 0)], (3, 2): [(2
, 2)], (0, 0): [(1, 0)], (1, 3): [(1, 2)], (2, 2): [(1, 2), (3, 2)], (3, 0): [(2
, 0)], (1, 0): [(0, 0), (2, 0), (1, 1)], (1, 1): [(1, 0), (1, 2)], (0, 2): [(1,
2)]}
```

Задание 2.3: написать функцию для нахождения выхода из лабиринта

```
def find_path(G, maze, node):
    if maze[node[0]][node[1]] == 'G':
        return [node]

    maze[node[0]][node[1]] = 'V'

    for i in G[node]:
        if maze[i[0]][i[1]] != 'V':
            path = find_path(G, maze, i)
            if path:
                return [node] + path
```