

Fecha: 28 Enero, 2018
Matricula: A00354513
Nombre: Irving Norehem Llamas Covarrubias
Curso: Sistemas inteligentes
Tarea: Búsqueda no Informada – BFS
Profesor: Dr. Abraham Aldaco Castélum

Descripción

Se solicito un programa que recorra un árbol construido mediante el algoritmo de BFS, para resolver el problema del 8-puzzle. El estado objetivo y que identifica como el estado a llegar es 0 1 2 3 4 5 6 7 8.

Instrucciones ejecutar programa

El programa es un script de python, con lo que simplemente hay que correrlo como **python bfs8puzzle.py**. El programa se ejecuto en una máquina virtual que tiene instalado Debian 8, con el siguiente intérprete de python: Python 2.7.13 :: Anaconda 4.4.0 (64-bit).

Código

```
import Queue
import time
import sys

possible_movements = [[2,4],[1,3,5],[2,6],[1,5,7],[2,4,6,8],[3,5,9],[4,8],[5,7,9],[6,8]]
direction = [['R','D'], ['L','R','D'], ['L','D'], ['U','R','D'], ['U','L','R','D'], ['U','L','D']]
empty_symbol = "0"
goal_state = "012345678"
file_name = "entrada.txt"

def generateNeighbors(state):
    list_result = []
    empty_position = state[0].find(empty_symbol)
    i = 0
    level = state[2]+1
    for position in possible_movements[empty_position]:
        tmp_result = state[0]
        tmp_result = tmp_result[0:empty_position] + tmp_result[position-1] + tmp_result[empty_position:]
        tmp_result = tmp_result[0:position-1] + empty_symbol + tmp_result[position:9]
        movement = direction[empty_position][i]
        i = i+1
```

```
list_result.append([tmp_result,state[1]+movement,level])
return list_result

def bfs( initialState , goalTest ):
    frontier = Queue.Queue()
    frontier.put(initialState)
    copy_frontier = set()
    copy_frontier.add(initialState[0])
    explored = set([])
    nodes_expanded = 0

    while not frontier.empty():
        state = frontier.get()
        copy_frontier.remove(state[0])
        explored.add(state[0])
        nodes_expanded = nodes_expanded + 1

        if state[0] == goal_state:
            return [state,nodes_expanded]

    neighbors = generateNeighbors(state)
    for neighbor in neighbors:
        if not( neighbor[0] in copy_frontier ) and not(neighbor[0] in explored) :
            frontier.put(neighbor)
            copy_frontier.add(neighbor[0])

    return False

# input_test = "724506831" "125340678" "813402765" "013425786"
file = open("entrada.txt", "r")
input_test = file.readline()
input_test = input_test[0:len(input_test)-1]
test_v = [input_test,"",0]

start = time.time()
result = bfs(test_v,goal_state)
end = time.time()

elapsed_time = end - start

if type(result) is list:
    number_nodes = result[1]
    result = result[0]
    print "Path to goal:\t\t" + str(result[1])
    print "Cost to the path:\t" + str(result[2])
    print "Nodes expanded/visited:\t" + str(number_nodes)
    print "Running time:\t\t" + str(elapsed_time) + " seconds"
```

```
print "Used memory:\t\t" + str(sys.getsizeof(result)*number_nodes) + " bytes"
else:
print str(result)
```

Casos de prueba

724506831

```
irvingnor@debian:~/Maestria/semestre_2$ python bfs_8puzzle.py
Path to goal:          LURDRDLLURRDLURRULLDRRULL
Cost to the path:      26
Nodes expanded/visited: 169741
Running time:          2.2316839695 seconds
Used memory:           16295136 bytes
```

125340678

```
irvingnor@debian: ~/Maestria/semestre_2
File Edit View Search Terminal Help
irvingnor@debian:~/Maestria/semestre_2$ python bfs_8puzzle.py
Path to goal:          ULL
Cost to the path:      3
Nodes expanded/visited: 10
Running time:          0.000663995742798 seconds
Used memory:           960 bytes
```

813402765

```
irvingnor@debian: ~/Maestria/semestre_2
File Edit View Search Terminal Help
irvingnor@debian:~/Maestria/semestre_2$ python bfs_8puzzle.py
Path to goal:          LURDLDRULURRDDLULULDRUL
Cost to the path:      22
Nodes expanded/visited: 92648
Running time:          1.13094305992 seconds
Used memory:           8894208 bytes
```

013425786

```
irvingnor@debian: ~/Maestria/semestre_2
File Edit View Search Terminal Help
irvingnor@debian:~/Maestria/semestre_2$ python bfs_8puzzle.py
Path to goal:          RDLDRUULDDLURRDLULURUL
Cost to the path:      22
Nodes expanded/visited: 80698
Running time:          1.20996904373 seconds
Used memory:           7747008 bytes
```