



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

<i>Profesor:</i>	Jorge Solano
<i>Asignatura:</i>	Estructura de Datos y Algoritmos II
<i>Grupo:</i>	2
<i>No de Práctica(s):</i>	7
<i>Integrante(s):</i>	Martínez Ostoa Néstor Iván
<i>Semestre:</i>	2019-1
<i>Fecha de entrega:</i>	28/Septiembre/2018
<i>Observaciones:</i>	

CALIFICACIÓN: _____

Práctica 5: Algoritmos de Grafos. Parte 1

Objetivo:

El estudiante conocerá las formas de representar un grafo e identificará las características necesarias para entender el algoritmo de búsqueda por expansión.

Desarrollo:

La idea de los grafos es prometer una eficiencia de búsqueda muy rápida, puesto que en cada iteración se cumplen condiciones que hacen el proceso más eficiente y por ende, una complejidad logarítmica.

El algoritmo de Breath First Search permite crear un árbol de expansión, dentro de un grado determinad, para un nodo raíz.

Código - implementación en Python

```
class Node:
    to = 0
    previous = None
    cost = 0
    distance = -1
    color = "White"

class Graph:
    edges = []
    grades = []
    numNodes = 0
    numEdges = 0
    isDirected = False
    hasCost = False
```

```

def mainAdyacentList():
    g = Graph()
    g.isDirected = True if int(input("Directed (1) ? Not Directed (2)? : ")) == 1 else False
    g.hasCost = True if int(input("Cost (1) ? No Cost (2)? : ")) == 1 else False
    MAXV = int(input("Number of nodes: "))
    s=(MAXV,MAXV)
    matrix = np.zeros(s)
    startGraph(g, MAXV)
    g.numNodes = MAXV
    g.numEdges = int(input("Number of edges: "))
    createGraph(g, matrix)
    print("Graph length: " + str(len(g.edges)))
    printGraph(g)
    BFS(g,3)

```

Resultados -> lista de adyacencia para la lista de WhatsApp

```

Directed (1) ? Not Directed (2)? : 2
Cost (1) ? No Cost (2)? : 1
Number of nodes: 20
Number of edges: 44
Graph length: 21
1 12 : 1 9 : 1 11 : 1 6 : 2 7 : 1 2 : 1 5 : 1
2 12 : 1 9 : 1 3 : 1 19 : 1 5 : 1 6 : 1 7 : 1 1 : 1
3 4 : 2 2 : 1
4 3 : 2 "White"
5 20 : 1 7 : 1 8 : 1 6 : 1 18 : 1 2 : 1 1 : 1
6 12 : 1 9 : 1 13 : 1 5 : 1 2 : 1 1 : 2
7 10 : 1 8 : 2 5 : 1 2 : 1 1 : 1
8 18 : 1 13 : 2 10 : 2 7 : 2 5 : 1
9 12 : 1 11 : 1 6 : 1 2 : 1 1 : 1
10 12 : 1 13 : 1 8 : 2 7 : 1
11 15 : 1 17 : 2 12 : 1 9 : 1 1 : 1
12 15 : 1 17 : 2 11 : 1 10 : 1 9 : 1 6 : 1 2 : 1 1 : 1
13 18 : 2 16 : 1 14 : 1 10 : 1 8 : 2 6 : 1
14 18 : 1 16 : 1 13 : 1
15 17 : 2 12 : 1 11 : 1
16 18 : 2 14 : 1 13 : 1
17 15 : 2 12 : 2 11 : 2
18 16 : 2 14 : 1 13 : 2 8 : 1 5 : 1
19 2 : 1
20 5 : 1

```

Resultados -> Árbol de expansión

Nodo raíz -> 3

```
1 - Black:2
2 - Black:1
3 - Black:0
4 - Black:1
5 - Black:2
6 - Black:2
7 - Black:2
8 - Black:3
9 - Black:2
10 - Black:3
11 - Black:3
12 - Black:2
13 - Black:3
14 - Black:4
15 - Black:3
16 - Black:4
17 - Black:3
18 - Black:3
19 - Black:2
20 - Black:3
```