

	<b>UNIVERSIDAD EAFIT</b> <b>ESCUELA DE INGENIERÍA</b> <b>DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS</b>	<b>Código: ST247</b>
		<b>Estructura de Datos 2</b>

## Laboratorio Nro. 3 Backtracking

**Santiago Soto Marulanda**  
Universidad Eafit  
Medellín, Colombia  
ssotom@eafit.edu.co

**Kevyn Santiago Gomez Patiño**  
Universidad Eafit  
Medellín, Colombia  
ksgomezp@eafit.edu.co

### 1) Codigos

8. Este algoritmo lo que hace es encontrar el camino más corto para esto lo que hace es, que dado un vértice inicial y empezara a tomar sus hijos e irá creando caminos con estos hijos peor a la vez comprueba con cuál tiene menos peso, hace eso hasta que todos los nodos estén visitados, para esta solución se apoyó de estructuras de datos como los ArrayList.

### 3) Simulacro de preguntas de sustentación de Proyectos

#### 1. Algoritmo voraz

En ciencias de la computación, un algoritmo voraz es una estrategia de búsqueda por la cual se sigue una heurística consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima. Este esquema algorítmico es el que menos dificultades plantea a la hora de diseñar y comprobar su funcionamiento. Normalmente se aplica a los problemas de optimización.

Tomado de: [https://es.wikipedia.org/wiki/Algoritmo\\_voraz](https://es.wikipedia.org/wiki/Algoritmo_voraz)

#### 2.

Fuerza Bruta		Vuelta Atrás	
Valor de N	Tiempo de Ejecución(ms)	Valor de N	Tiempo de Ejecución(ms)
4	0	4	0
5	0,07	5	0
6	1,12	6	0
7	26,81	7	0
8	544,11	8	0
9	13397,28	9	0
10	371598	10	0

**DOCENTE MAURICIO TORO BERMÚDEZ**  
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627  
Correo: mtorobe@eafit.edu.co

	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST247
		Estructura de Datos 2

11	Demasiado	11	0
12	Demasiado	12	0
13	Demasiado	13	0,1
14	Demasiado	14	0,6
15	Demasiado	15	0,4
16	Demasiado	16	4,1
17	Demasiado	17	2,4
18	Demasiado	18	23,3
19	Demasiado	19	1,3
20	Demasiado	20	109,3
21	Demasiado	21	5,3
22	Demasiado	22	1086,1
23	Demasiado	23	18,9
24	Demasiado	24	295,7
25	Demasiado	25	35,3
26	Demasiado	26	309,2
27	Demasiado	27	364,3
28	Demasiado	28	259,1
29	Demasiado	29	1390,9
30	Demasiado	30	58909,1
31	Demasiado	31	73249
32	Demasiado	32	99375,5
33	Demasiado	33	Demasiado
N	$O(N!)$	N	$O(N!)$

3. Esto depende de las necesidades de cada problema, si lo que se necesita es conocer los hijos de los nodos, o los hermanos de los nodos, en un ejemplo como encontrar el camino más corto entre dos vértices con BFS tendrá mayor complejidad pero dará la mejor solución, por el contrario con DFS tendrá menor complejidad pero no dará la mejor solución.

	<p style="text-align: center;">UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS</p>	<p>Código: ST247</p> <p>Estructura de Datos 2</p>
--	---------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------

#### 4. Algoritmo de Bloques

El algoritmo de bloques de Jungnickel es una adaptación del algoritmo DFS de Robert Tarjan, enunciado en el año 72, y que a su vez es una variación ligeramente más eficiente del algoritmo de Kosaraju. Permite hacer una partición de un grafo en los bloques de que está formado. Además, encuentra los puntos de corte del grafo original.

#### Algoritmo de Dijkstra

El algoritmo de Dijkstra debe su nombre a su creador, Edsger Dijkstra, quien lo concibió en 1956 (fue posteriormente publicado en 1959). Se trata de un algoritmo de búsqueda sobre grafos que soluciona el problema del camino mínimo en grafos de aristas con pesos no negativos.

Tomado de: <http://www.dma.fi.upm.es/personal/gregorio/grafos/web/iagraph/busqueda.html>

5. consideramos que no es posible utilizar backtracking para hallar el camino de menor costo entonces utilizamos dfs para hallar el camino desde el vértice de partida hasta el objetivo y retornamos un ArrayList con el camino.

6. Es  $O(n!)$

7. n es el número de nodos del grafo.

#### 4) Simulacro de Parcial

1)

$n-a, a, b, c$   
 $res, solucionar(n, c, a, b)$   
 $res, solucionar(n, b, c, a)$

2) línea 02  $graph.length$

línea 09  $v, graph, path, pos$

línea 11  $graph, path, pos+1$

3)a)

0->0,3,7,4,2,1,5,6

1->1,0,3,7,4,2,6,5

2->2,1,0,3,7,4,5,6

3->3,7

4->4,2,1,0,3,7,5,6

5->5

6->6,2,1,0,3,7,4,5

7->7

b)

0->0,3,4,7,2,1,6,5

1->1,0,2,5,3,4,7,6

2->2,1,4,6,0,5,3,7

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co

	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST247
		Estructura de Datos 2

3->3,7  
4->4,2,1,6,0,5,3,7  
5->5  
6->6,2,1,4,0,5,3,7  
7->7

```

4)
public static ArrayList<Integer> camino(Digraph g, int inicio, int fin) {
    boolean[] visitados = new boolean[g.size()];
    ArrayList<Integer> list = new ArrayList<>();
    if (dfs(g, inicio, fin, visitados, list))
        return list;
    return new ArrayList<Integer>();
}

private static boolean dfs(Digraph g, int nodo, int objetivo, boolean[] visitados, ArrayList<Integer> list)
{
    if(nodo == objetivo){
        list.add(nodo);
        return true;
    }else{
        ArrayList<Integer> successors = g.getSuccessors(nodo);
        visitados[nodo] = true;
        if(successors != null){
            for(int successor: successors){
                if(!visitados[successor]){
                    list.add(nodo);
                    return dfs(g, successor, objetivo, visitados, list);
                }
            }
            list.remove(list.size());
        }
    }
    return false;
}

```

5. Línea 7 1  
Línea 11 ni, nj

## 6) Trabajo en Equipo y Progreso Gradual


	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST247
		Estructura de Datos 2

a) Actas de reunión


Integrante	Fecha	Hecho	Haciendo	Por Hacer
Santiago Soto	07/03/18		Ejercicio 1.1 y 1.2	Ejercicio 1.3 y 1.4
Santiago Soto	07/03/18	Ejercicio 1.1 y 1.2	Ejercicio 1.3 y 1.4	Ejercicio 1.5
Santiago Soto	07/03/18	Ejercicio 1.3 y 1.4	Ejercicio 1.5	Ejercicio 3.1
Santiago Soto	07/03/18	Ejercicio 1.5 1.6 1.7 1.8	Ejercicio 3.1	Ejercicio 3.2 3.3
Santiago Soto	08/03/18	Ejercicio 3.2 3.3		
Kevyn Gómez	10/03/18		Ejercicio 2.1	Simulacro de parcial
Kevyn Gómez	11/03/18	Ejercicio 2.1	Simulacro de parcial	

b) El reporte de cambios en el código


History for [ST0247-032](#) / [laboratorios](#) / [lab03](#)

 Commits on Mar 11, 2018


**SubiendoCodigo**

 ssotom committed a minute ago

**Modificado ejercicio en linea**

 ksgomez committed 3 hours ago

**Haciendo ejercicio en linea**

 ksgomez committed 5 hours ago

c) El reporte de cambios del informe de laboratorio

	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST247
		Estructura de Datos 2

Hoy
▶ 11 de marzo, 21:43 Versión actual ■ Santiago Soto
▶ 11 de marzo, 20:39 ■ Kevyn S19
▶ 11 de marzo, 12:26 ■ Santiago Soto
viernes
▶ 9 de marzo, 10:47 ■ Santiago Soto

DOCENTE MAURICIO TORO BERMÚDEZ  
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627  
Correo: mtorobe@eafit.edu.co