

Laboratorio Nro. 3 Backtracking

David Alzate Cardona
Universidad Eafit
Medellín, Colombia
Dalzatec1@eafit.edu.co

Laura Alzate Madrid
Universidad Eafit
Medellín, Colombia
lalzatem@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Algoritmo de Dijkstra: resuelve el problema de los caminos más cortos desde un único vértice origen hasta todos los otros vértices del grafo.

Algoritmo de Bellman – Ford: resuelve el problema de los caminos más cortos desde un origen si la ponderación de las aristas es negativa.

Algoritmo de Búsqueda A*: resuelve el problema de los caminos más cortos entre un par de vértices usando la heurística para intentar agilizar la búsqueda.

Algoritmo de Floyd – Warshall: resuelve el problema de los caminos más cortos entre todos los vértices.

Algoritmo de Johnson: resuelve el problema de los caminos más cortos entre todos los vértices y puede ser más rápido que el de Floyd-Warshall en grafos de baja densidad.

Algoritmo de Viterbi: resuelve el problema del camino estocástico más corto con un peso probabilístico adicional en cada vértice.

3.2 hay $n(n-1)/2$ caminos

3.3

VALOR DE N	TIEMPO EJECUCIÓN FUERZA BRUTA	TIEMPO EJECUCIÓN BACKTRAKING
4	0,196s	0,163s
5	0,152s	0,174s
6	0,149s	0,172s
7	0,142s	0,172s
8	0,159s	0,171s
9	0,148s	0,180s
10	0,142s	0,173s
11	0,165s	0,188s
12	0,309s	0,207s
13	1,083s	0,233s
14	5,988s	0,181s

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 2

Código ST0247

15	39,484s	0,182s
16	4m50,081s	0,231s
17	Demora mucho	0,209s
18	Demora mucho	0,268s
19	Demora mucho	0,223s
20	Demora mucho	0,259s
21	Demora mucho	0,203s
22	Demora mucho	0,868s
23	Demora mucho	0,212s
24	Demora mucho	0,380s
25	Demora mucho	0,290s
26	Demora mucho	0,455s
27	Demora mucho	0,471s
28	Demora mucho	1,756s
29	Demora mucho	1,014s
30	Demora mucho	31,664s
31	Demora mucho	7,849s
32	Demora mucho	54,537s
N	O(n!)	O(2^n)

3.4 DFS se utiliza para encontrar el camino más óptimo de un punto a otro o para recorrer todos los puntos de un grafo volviendo al inicio, mientras que BFS se utiliza para problemas como el colorear un mapa con el mínimo de colores o saber si es posible pintar un mapa con n colores.

3.5 la estructura de datos es arraylist, ya que esta de manera dinámica puede eliminar y guardar datos de forma dinámica sin decir un tamaño, el programa recibe un primer parámetro en el cual se compone de dos cosas(N,M), N es el primer valor, es el número de nodos y el segundo (M) es el número de vértices, luego se empieza a enviar el número de arcos de forma que el primer valor es el nodo origen, el segundo valor es el nodo de llegada y el tercero es el peso (todos estos se digitan a la misma vez separados por un espacio ya que se utiliza un Split para separar cada uno), luego de esto el programa empieza a calcular cual es el mejor recorrido para llegar del punto inicial al punto N por medio de recorrido de grafos en profundidad.

3.8. Este algoritmo funciona por medio del recorrido en profundidad, con la ayuda de backtracking. Lo primero es recorrer el grafo con dfs desde un nodo inicial a un nodo final, y a la vez ir acumulando el costo o peso del recorrido, al finalizar cada posible recorrido se va a guardar el valor final en una variable y se ira preguntando cual es la menor, para al final poder retornar el menor costo

4) Simulacro de Parcial

4.1 A) $(n-a,a,b,c)$

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

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473



ESTRUCTURA DE DATOS 2
Código ST0247

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

4.2 a) graph.length

b) (v, graph, path, pos)

c) (graph, path, pos+1)

4.3 A) 0 -> 3 -> 7 -> 4 -> 2 -> 1 -> 5 -> 6

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

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

3 -> 7

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

5

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

7

B) 0 -> 3 -> 4 -> 7 -> 2 -> 1 -> 6 -> 5

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

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

3 -> 7

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

5

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

7

4.4 -

4.5 A) 1

B) n_i, n_j .

C) $O(2 \text{ a la } N)$

4.6 1) c.

2) a.

4.7 A) $N == R$

B) i

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 2
Código ST0247

C) $r+1$

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

