

Laboratorio Nro. 3

Vuelta atrás (Backtracking)

Jacobo Rave Londoño
Universidad Eafit
Medellín, Colombia
jravel@eafit.edu.co

Diego Alejandro Vanegas González
Universidad Eafit
Medellín, Colombia
davanegas@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Existen otras técnicas utilizadas para encontrar el camino o recorrido más corto en un grafo, estas son algunas: Dijkstra algorithm, Bellman Ford Algorithm, A* Search Algorithm, Warshall Algorithm, Johnson Algorithm, Viterbi Algorithm. Cada uno tiene una buena efectividad dependiendo de las condiciones en que se encuentre el grafo. De lo contrario el algoritmo puede presentar fallas o no terminar la búsqueda.

3.2 Para calcular los posibles caminos de un grafo para encontrar el camino más corto se puede hacer una sumatoria donde se divida la cantidad factorial de vértices entre la cantidad de vértices menos uno factorial. Así cada camino se cuenta en un grafo completamente dirigido.

3.3

	Backtraking	Brute force			
			13	0	213
1	0	0	14	1	1308
2	0	0	15	1	8540
3	0	0	16	1	34238
4	0	0	17	5	234523
5	0	0	18	2	más de 50 min
6	0	0	19	13	más de 50 min
7	0	0	20	0	más de 50 min
8	0	1			
9	1	1			
10	0	3			
11	0	9			
12	0	40			

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

21	68	más de 50 min
22	3	más de 50 min
23	732	más de 50 min
24	11	más de 50 min
25	193	más de 50 min
26	24	más de 50 min
27	199	más de 50 min
28	237	más de 50 min
29	1682	más de 50 min
30	899	más de 50 min
31	26108	más de 50 min
32	5609	más de 50 min

3.4 Es conveniente utilizar DFS si se quiere encontrar a profundidad el primer camino o la primera solución que surja de un problema, la ventaja es que casi no consume memoria, pero es menos efectivo a la hora de resolver problemas. En cambio, al utilizar BFS es porque queremos recorrer todo el grafo y mirar en un posible ejercicio cual es el camino más corto o también colorear grafos para encontrar cuantos colores utilizar, este necesita más memoria pero es más efectivo a la hora de resolver problemas.

3.5 La estructura de datos que utilizamos para resolver este problema fue grafos con listas de Adyacencia. Está es la mejor forma de representar el grafico con estas condiciones. La forma para hallar el mejor camino, es usando Backtracking. Comenzamos con un nodo Source, miramos sus sucesores y entonces comenzamos a recorrer cada posible ruta que logre llegar al destino.

3.6 La complejidad es $O(n+m)$

3.7 n es el número de vértices y m el número de arcos

3.8 utilizamos bfs para recorrer todos los caminos del grafo y cada vez que se recorre el grafo se van sumando el peso de cada vértice del camino y se va comparando con el anterior para retornar el camino más corto entre estos.

4) Simulacro de Parcial

- 4.1** $n-a, a, b, c$
 res, solucionar($n-b, a, b, c$) +1
 res, solucionar($n-c, a, b, c$) +1
- 4.2** $pos == graph.length$
 sePuede($v, graph, path, pos$)
 cicloHamilAux($graph, path, pos +1$)
- 4.3** DFS = 0,3,7,4,2,6,1,5
 BFS = 0,3,4,7,2,1,6,5

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

4.4

4.5 Return $1 + \text{lcs}(i - 1, j - 1, s1, s2)$
Math.max(nj, ni)

4.6 C

a

4.7 $r \geq 8$;
a[r] = 1;
sol(a, 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

