

Laboratorio Nro. 3 Backtracking

Juan Camilo Jiménez Rojas
Universidad Eafit
Medellín, Colombia
jcjimenezr@eafit.edu.co

Santiago Espinosa Valderrama
Universidad Eafit
Medellín, Colombia
sespinosav@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Algunos algoritmos relacionados para encontrar el camino más corto en un gráfico son dijkstra, el algoritmo de bellman ford y el algoritmo de floyd.

3.2 $n(n-1)/2$ vertices

3.3

Reinas	Fuerza Bruta
4	2ms
6	3ms
8	7ms
...	...
32	5 min 27 seg

Reinas	Backtracking
4	0ms
6	1ms
8	1ms
...	...
32	1 min 32 seg

3.4 Se utiliza BFS cuando necesitamos conocer todas las posibles formas entre un vértice en específico y sus sucesores, verificando camino más corto entre los nodos, evaluando y el cada uno con el costo más bajo y selecciona.

Por otro lado, usamos DFS para verificar la ruta más corta entre dos nodos, este algoritmo toma un vértice y luego sucesor, siempre elige el nodo más profundo para expandirlo dando como resultado una evaluación más rápida.

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

3.5 La estructura de datos utilizada en este problema es bracktracking, porque con esto podemos encontrar la forma más corta entre el vértice inicial y el final sin la necesidad de pasar por todos los vértices, el objetivo es alcanzar el destino en el costo mínimo.

3.6 $O(n^2)$

3.7 N es un grafo el cual contiene los sucesores.

3.8 El algoritmo se encarga de evaluar el peso del camino de un nodo a otro con backtracking. Consiste trata de una recursion que va a cada sucesor de cada nodo adyacente al nodo anterior hasta llegar al nodo de destino y retorna cada peso de los caminos sumando los pesos de los arcos que aquí participan, evaluando todos los caminos existentes y retorna el peso del camino más favorable.

4) Simulacro de Parcial

4.1

4.1.1: $n-a, a, b, c$

4.1.2: $res, solucionar(n-b, a, b, c) + 1$

4.1.3: $res, solucionar(n-c, a, b, c) + 1$

4.2

4.2.1: $pos == path[0]$

4.2.2: $v, graph, path, pos$

4.2.3: $graph, path, v$

4.3

4.3.1:

desde 0: 0,3,4,7,2,1,6,5

desde 1: 1,0,2,5,3,4,6,7

desde 2: 2,1,4,6,0,5,3,7

desde 3: 3,7

desde 4: 4,2,1,6,0,5,3,7

desde 5: 5

desde 6: 6,2,1,4,0,5,3,7

desde 7: 7

4.3.2:

desde 0: 0,3,7,4,2,1,5,6

desde 1: 1,0,3,7,4,2,6,5

desde 2: 2,1,0,3,7,4, 5,6

desde 3: 3,7

desde 4: 4,2,1,0,3,7,5,6

desde 5: 5

desde 6: 6,2,1,0,3,7,4,5

desde 7: 7

4.4

4.5

4.5.1. Linea7: 1

4.5.2 Linea 11: n_i, n_j

4.5.3 $O(2^n)$

4.6

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

4.6.1 *c*

4.6.2 *a*

4.7

4.7.1 *Línea 3: $N=r$*

4.7.2 *Línea 8: i*

4.7.3 *Línea 9: $r+1$*

6) Trabajo en Equipo y Progreso Gradual (Opcional)

Responsable	Fecha	Listo	Por hacer
Camilo / Santiago	14/03/19	Entender punto 1	Buscar solución
Santiago	15/03/19	Implementar y probar solución	Leer punto 2
Camilo / Santiago	16/03/19	Entender punto 2	Buscar solución
Santiago	17/03/19	Implementar y probar solución	Subir laboratorio
Camilo	18/03/19	Terminar y subir laboratorio	

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