

Laboratorio Nro. 3

Vuelta atrás (*Backtracking*)

Agustín Rico Piedrahita
Universidad Eafit
Medellín, Colombia
aricop@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Se pueden utilizar todas las técnicas de recorridos de grafos, como BFS o DFS.

3.2 En el peor de los casos (grafo completo) hay $(V - 2)! \sum_{k=0}^{V-2} \frac{1}{(V-2-k)!}$ caminos posibles, con $V =$ número de nodos.

3.3

3.4 En árboles/grafos muy amplios (con muchos nodos a un mismo nivel de profundidad), el recorrido con BFS puede ser lento y consumirá mucha memoria. Sin embargo, si las soluciones/nodos objetivos son muy poco frecuentes, el recorrido usando BFS podría ser mejor.

3.5 Para el problema 2.1 se utilizó una versión modificada del recorrido BFS. Por lo tanto, se utilizó la estructura de datos de un árbol, el cual utiliza un diccionario que contiene todos los nodos que pertenecen a cada nivel de profundidad. La implementación de la clase Árbol implica una implementación de la clase *Nodo*.

3.6 En esta implementación de BFS, en el peor de los casos cada vértice se visita una vez, sin embargo, los nodos se pueden visitar hasta $(V - 1)!$ veces. Por lo tanto, la complejidad del algoritmo es de $O((V - 1)! + E)$

3.7 V es el número de vértices y E es el número de arcos

3.8 El algoritmo del numeral 1.1 es el mismo que el del 2.1, que consiste en una versión modificada de BFS, con la diferencia que los nodos ya visitados puede volver a ser visitado si no pertenece a un nivel de profundidad superior en el árbol. De esta manera, se exploran todos los caminos posibles al nodo destino y se elige la de menor distancia.

4) Simulacro de Parcial

4.1

4.1.1 *solucionar*($n - a, a, b, c$)

4.1.2 *solucionar*(*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

4.1.3 $\text{solucionar}(\text{res}, \text{solucionar}(n - c, a, b, c))$

4.2

4.2.1 graph.length

4.2.2 $(v, g, \text{path}, \text{pos})$

4.2.3 $(g, \text{path}, \text{pos} + 1)$

4.3

4.3.1

Iniciando en:	Orden:
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	6,2,1,0,3,7,5,4
7	7

4.3.2

Iniciando en:	Orden:
0	0,3,4,1,2,7,5,6
1	1,0,5,3,4,7,2,6
2	2,1,4,6,0,5,3,7
3	3,7
4	0,2,3,1,6,7,5
5	5
6	2,1,4,0,5,3,7
7	7

4.4

4.5

4.5.1 $s1.\text{chartAt}(i)$

4.5.2 n_i, n_j

4.5.3 n

4.6 a

4.6.1 $c)$

4.6.2 $a)$

4.7 A

4.7.1 $N == r$

4.7.2 i

4.7.3 $r + 1$

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