

Laboratorio Nro. 3 Vueltas atrás (Backtracking)

Miguel Ángel Martínez Florez
Universidad Eafit
Medellín, Colombia
mamartinef@eafit.edu.co

Pablo Micolta López
Universidad Eafit
Medellín, Colombia
pmicolta@eafit.edu.co

Daniel Ricardo Palacios Diego
Universidad Eafit
Medellín, Colombia
drpalaci@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1

R/ En el problema del camino más corto también se pueden usar algoritmos Greedy, tales como A* o Bellman-Ford.

3.2

R/ Asumiendo que no vamos a repetir ningún vértice, podemos asumir que van a ver n factorial que son todos los caminos que usa la totalidad de los vértices y luego se les suman $n-1!$, $n-2!$, $n-3!$, $n-4!$, ..., $n-m!$; siendo m la cantidad de vértices que no se usaron en el recorrido.

3.3

R/

| N | tiempo |
|----|----------|
| 4 | 0.001004 |
| 5 | 0.003404 |
| 6 | 0.022627 |
| 7 | 0.075972 |
| 8 | 0.581994 |
| 9 | 5.689046 |
| 10 | 66.41700 |
| N | $O(N!)$ |

3.4

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

R/ El algoritmo DFS se utiliza para problemas donde hay cierta dependencia de sus antecesores o cuando se necesita que la progresión de la solución cumpla con algún parámetro; por otro lado, tenemos el Algoritmo BFS que se utiliza mayormente en los problemas que requieren elegir el mejor camino posible.

3.5

R/ utilizamos listas doblemente enlazadas en un grafo para representar las aristas entre los vértices y como no todos los vértices se conectaban entre sí, observamos que era más eficiente utilizar listas enlazadas para la memoria. Además, utilizamos un arreglo para representar los ya visitados, ya que, nos da un acceso más rápido al valor de cada posición.

3.6

R/ la complejidad es $O(n^2 \cdot m)$, porque en la recursión cada vez que hacemos un paso, estamos eliminando un vértice que se tiene que revisar y eso en $O(n^2)$ y eso se hace una vez para cada arista que tiene el nodo, que en el peor de los casos son todas las aristas (m).

3.7

R/ n es la cantidad de vértices y m es a cantidad de aristas.

3.8

R/ Utilizamos un diccionario que usaba las id dadas para representar las conexiones del grafo; luego recorrimos por longitud, iniciando desde un nodo dado hasta el nodo objetivo y cuando se llegaba a este objetivo se creaba una lista enlazada a la cual se le añadía el nodo final y se le inicializaba la distancia en 0. Finalmente, se añadía la distancia que se iba acumulando con la distancia entre el nodo actual y uno de los sucesores y si era menor a la distancia mas pequeña que se tenía hasta ahora, se actualizaba tanto la distancia como la lista acordeamente.

4) Simulacro de Parcial

4.1 .1 $n-a, b, c$

4.1.2 $res, solucionar(n, b, c, a)$

4.1.3 $res, solucionar(n, c, a, b)$

4.2.1 $path.length()$

4.2.2 $v, graph, path, pos$

4.2.3 $graph, path, v$

4.5.1 1

4.5.2 n_i, n_j

4.5.3 n^2

4.7.1 $r == N$

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



ESTRUCTURA DE DATOS 2
Código ST0247

5) Lectura recomendada (opcional)

Mapa conceptual

6) Trabajo en Equipo y Progreso Gradual (Opcional)

6.1 *Actas de reunión*

6.2 *El reporte de cambios en el código*

6.3 *El reporte de cambios del informe de 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

