

Laboratorio Nro. 3

Vuelta atrás (Backtracking)

Sara María Castrillón Ríos
Universidad Eafit
Medellín, Colombia
smcastril1@eafit.edu.co

Valeria Guerra Zapata
Universidad Eafit
Medellín, Colombia
vguerraz@eafit.edu.co

1. ____
2. ____

3. Simulacro de preguntas de sustentación de Proyectos

3.1. Tenemos varios algoritmos para este problema:

- 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. Infinitos

3.3. Tiempos de ejecución con diferentes valores n para el programa de las n reinas realizado en el numeral 1.2

Valor de N	Tiempo de ejecución
4	0.0004229545593261719
5	0.00042748451232910156
6	0.0065000057220458984
8	0.05543828010559082
10	0.09262800216674805
15	2.0683047771453857
20	714.5665466785431

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

25	315.238338470459
N	O(N!)

3.4. DFS los usamos para:

- Encontrar nodos conectados en un grafo
- Ordenamiento topológico en un grafo acíclico dirigido
- Encontrar puentes en un grafo de nodos
- Resolver puzzles con una sola solución, como los laberintos
- Encontrar nodos fuertemente conectados

BFS lo usamos para:

- Encontrar el camino más corto entre 2 nodos, medido por el número de nodos conectados
- Probar si un grafo de nodos es bipartito (si se puede dividir en 2 conjuntos)
- Encontrar el árbol de expansión mínima en un grafo no ponderado
- Hacer un Web Crawler
- Sistemas de navegación GPS, para encontrar localizaciones vecinas

3.5. Para este ejercicio se utilizaron listas, también se utilizó la técnica de backtracking, en donde cada llamado recursivo agregaba un elemento al arreglo path (donde estará el camino más corto) la recursión finaliza cuando el último elemento de la lista path es igual a nuestro destino (retorna path). Para lograr esto se utilizó además una lista visitados que nos asegurara que el elemento que se desea agregar no haya sido agregado antes.

3.6. Complejidad 2.1

3.7. Significado de las variables de la complejidad

3.8. Explicación 1.1 implementación, funcionamiento, estructuras de datos y algoritmos usados

4. Simulacro de Parcial

4.1. ____

4.2. Respuestas:

4.2.1. `path.length // g.length`

4.2.2. `v, graph, path, pos`

4.2.3. `Graph, path, pos+1`

4.3. ____

4.4. ____

4.5. Respuestas:

4.5.1. `S1.charAt(i)`

4.5.2. ..

4.5.3. ..

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

4.7. Respuestas:

4.7.1. „

4.7.2. ,

4.7.3. ,

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

