

	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST245
		Estructura de Datos 1

Laboratorio Nro. 3: Vuelta atrás (Backtracking)

Isaias Labrador Sanchez
 Universidad Eafit
 Medellín, Colombia
 ilabradors@eafit.edu.co

Santiago Hincapié Murillo
 Universidad Eafit
 Medellín, Colombia
 shincapiem@eafit.edu.co

Andrés Almanzar Restrepo
 Universidad Eafit
 Medellín, Colombia
 aalmanzarr@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Una de las técnicas computacionales es el uso del “Algoritmo A*” siendo este un algoritmo Heurístico, etiqueta los nodos y determina la probabilidad de dichos nodos de pertenecer al camino más optimo.

Otra técnica usada para resolver el problema del camino más corto es el “Algoritmo del vecino más próximo”. Dicho algoritmo consta de pocos pasos. Se tomará un nodo al azar y desde allí se buscará el nodo más cercano. Luego se cambiará el nodo anterior por el nuevo y se llamará el nodo actual. De esta manera se buscará el nodo deseado. Todo este proceso se lleva a cabo sin repetir nodos puesto que si se hiciera llevaría a un bucle infinito.

3.3 Generalmente se utiliza Breadth first search o BFS cuando se desea encontrar el camino más corto desde un nodo origen a otro destino. Esto induce a que sería mucho más optimo representar un algoritmo como les el algoritmo de Dijkstra con un BFS.

Por otro lado el uso de Depth first search o DFS es más conveniente para el uso de algoritmo que utilicen la fuerza bruta. Esto porque se revisa todas las posibles soluciones.

Ambos tipos de búsquedas son útiles para encontrar si hay conexión entre un nodo y otro. Pero un uso más adecuado sería para un problema en específico.

3.4 Algoritmo A*: Este algoritmo es uno de los más conocidos además de BFS y DFS. Consiste en los siguientes pasos:

1. Elegir un vértice de forma aleatoria.
2. Elegir una arista vecina al nodo actual. (arista vecina de menor peso)
3. Ahora la nueva arista será el nodo actual.
4. Marcar el nodo actual como visitado.
5. Ir de esta manera hasta encontrar el nodo destino, en otras palabras volver al paso 2. (No se puede ir a un nodo ya visitado)
6. Si todos los vértices están visitados, terminar algoritmo.

3.7 Para calcular la complejidad se trata de n que se refiere a el tamaño del stack de nodos que utilizamos y sacar de allí los vecinos de los nodos almacenados en él, m se refiere a cada uno de los vecinos del nodo que se le hace pop en el stack, los cuales se utilizan para decidir si el mismo ya ha sido visitado y así añadido o no al stack y generar una reacción en cadena, que solo termina cuando todos los nodos han sido visitados.

DOCENTE MAURICIO TORO BERMÚDEZ
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627
Correo: mtorobe@eafit.edu.co

	UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS	Código: ST245
		Estructura de Datos 1

4) Simulacro de Parcial

1. a) (n, a) ;
b) (a, b) ;
c) (a, c) ;
2. a) 0;
b) $(v, \text{path}[], \text{graph}[], \text{pos})$;
c) $(\text{graph}[], \text{path}[], v)$;
3. a) 0,4,2,6,1,5,3,7
b) 0,3,7,4,2,6,1,5
5. 5.1 $\text{lcs}(i, i, s1, s2)$;
5.2 n_i, n_j
5.3 $O(n)$
6. 6.1) c 0,1,4,2,3
6.2) c 0,4,3,2,1