

Laboratorio Nro. 5: Programación Dinámica

Alejandro Cano Munera

Universidad Eafit
Medellín, Colombia
acanom@eafit.edu.co

Sebastián Giraldo Gómez

Universidad Eafit
Medellín, Colombia
sgiraldog@eafit.edu.co

3) Simulacro de preguntas de sustentación de proyecto:

3.1) Para la solución del agente viajero con programación dinámica o algoritmo de Held-Karp se utilizan dos tablas donde se guardan los caminos entre los diferentes subconjuntos de nodos y los subconjuntos de nodos disponibles, las tablas son representadas con matrices, y el manejo del grafo propiamente se hace a través de listas enlazadas. La función principal de las tablas es guardar los cálculos de los caminos entre los diferentes nodos para posteriormente acceder a ellos más rápidamente, reduciendo el tiempo de ejecución y el costo de las operaciones. El camino más corto se logra al comparar las posibilidades generadas por todos los subconjuntos que se van creando en un grafo, luego de hacer esta comparación, se guarda la mejor opción y esto se repite hasta terminar el ciclo, es decir, haber recorrido todos los nodos del grafo.

3.2) Para resolver el problema del agente viajero aparte de programación dinámica y algoritmos voraces también existe la “Fuerza bruta” que básicamente consiste en comparar la distancia del nodo actual con todos los otros nodos del mapa, para así hallar el mejor camino. Aunque se corrobora que el camino elegido es el más corto de todos (ya que se comparan todas las otras posibles), el algoritmo es muy lento, ya que tiene que realizar muchas operaciones para resolver el problema; su complejidad es $O(n!)$.

3.3) Para resolver este problema se usaron arreglos de dos posiciones en un ArrayList para tener un arreglo con las posiciones de los desechos, y al tener esto se utilizó la fórmula de distancia entre dos puntos para determinar cuál era la menor distancia y así hacer los movimientos, al final se hacía la suma del recorrido restando las “x” y las “y” de los movimientos para así tener la longitud del movimiento lineal y no diagonal.

3.4) $T(n) = n^2 * 2^n + C$ es $O(n^2 * 2^n)$

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: mtorobe@eafit.edu.co

3.5) En este ejercicio n es el número de nodos que posee el grafo dirigido que se está evaluando en el problema.

4) Simulacro de parcial:

1.

1.1

	c a l l e					
	0	1	2	3	4	5
c	1	0	1	2	3	4
a	2	1	0	1	2	3
s	3	2	1	1	2	3
a	4	3	2	2	2	3

1.2

	m a d r e					
	0	1	2	3	4	5
m	1	0	1	2	3	4
a	2	1	0	1	2	3
m	3	2	1	1	2	3
a	4	3	2	2	2	3

2.

2.1 $O(md)$ m: longitud cadena 1, d: longitud cadena 2.

2.2 return table[lenx][leny];

3.

3.1 a) $O(n)$

3.2 a) $T(n) = c1:n + c2$

4. c) $O(2^n)$ y se optimiza con programación dinámica.