David Martínez Díaz 44669141 J

- Ejercicio 1:

El problema del camino mínimo, se trata de un grafo $G=(N,A)$, donde N es el conjunto de nodos y A el conjunto de sus arcos. Donde cada arco tiene asociada una longitud no negativa.

⇒ Su principal objetivo es obtener el camino de longitud mínima para cualquier par de nodos.

⇒ Para ello utilizamos una matriz "L" donde guardemos las longitudes de los arcos, los cuales estén enumerados desde 1 hasta n y donde se diferencian distintos casos.

$L(i,j)=0$ / $L(i,j) \geq 0$ si $i \neq j$ / $L(i,j)=\infty$ si no existe.

➡ Primera característica: n-etápico.

Para poder aplicar Programación Dinámica, el problema debe ser n-etápico, en este caso SI lo es, ya que para encontrar el camino desde un vértice i al j, vamos etapa por etapa para saber cual va a ser el segundo nodo, mas adelante el 3º, etc. hasta alcanzar j.

➡ Segunda característica: Principio del Óptimo de Bellman

Para hacer la comprobación, imaginemonos un camino:
{Málaga - Valencia - Madrid - Barcelona - Marsella}, como el camino más corto desde Málaga a Marsella.

⇒ Si avanzamos a la siguiente etapa, es decir, Valencia, el camino optimo sería {Valencia - Madrid - Barcelona - Marsella} para llegar a Marsella, porque si no:

David Martínez Díaz 44669141J

- Si en el caso de que el camino {Valencia - Toledo - Plasencia - Marsella} sea el camino más corto a Marsella.

- Entonces el camino {Malaga - Valencia - Madrid - Barcelona - Marsella} sería un camino pero no sería el más optimo.

⇒ Como se produce una contradicción, por tanto se verifica el POB.

Por último, al crear su ecuación recurrente, tenemos A_i como el conjunto de los vértices adyacentes al vertice i.

Además, tenemos que sea I_k el camino mínimo desde k hasta j es el camino más corto de los caminos del conjunto.

Tenemos esta ecuación:

$$D_k(i,j) = \min\{D_{k-1}(i,j), D_{k-1}(i,k) + D_{k-1}(k,j)\}.$$

David Martínez Driez 44669141J

- Ejercicio 2: Como mi DNI termina en 1 ⇒ $\boxed{A=8}$

$$\begin{bmatrix} 0 & 5 & 1 & 8 \\ 5 & 0 & 4 & 5 \\ 8 & 4 & 0 & 6 \\ 8 & 5 & 6 & 0 \end{bmatrix}$$

1º En primer lugar calculamos las distancias directas con el origen:

$$\begin{cases} \cdot\, g(2, \emptyset) = 5 \\ \cdot\, g(3, \emptyset) = 8 \\ \cdot\, g(4, \emptyset) = 8 \end{cases}$$

2º. Lo siguiente es calcular la distancia, pero con un nodo en el camino a través de la fórmula:

⇒ Para el nodo $i=2$:

$g(2, \{3\}) \Rightarrow L_{23} + g(3, \emptyset) = 4 + 8 = 12$

$g(2, \{4\}) \Rightarrow L_{24} + g(4, \emptyset) = 5 + 8 = 13$

⇒ Para el nodo $i=3$:

$g(3, \{2\}) \Rightarrow L_{32} + g(2, \emptyset) = 4 + 5 = 9$

$g(3, \{4\}) \Rightarrow L_{34} + g(4, \emptyset) = 6 + 8 = 14$

⇒ Para el nodo $i=4$:

$g(4, \{2\}) = L_{42} + g(2, \emptyset) = 5 + 5 = 10$

$g(4, \{3\}) = L_{43} + g(3, \emptyset) = 6 + 8 = 14$

3º Procedemos a buscar el camino pero ahora con 2 nodos:

→ Para el conjunto $\{2,3\}$:

$g(4, \{2,3\}) \Rightarrow \text{Min}\,[L_{42} + g(2, \{3\}),\ L_{43} + g(3, \{2\})]$

$\hookrightarrow [5+12,\ 6+9] = [17, 15] = 15$

Escaneado con CamScanner

David Martínez Díaz 44669191S

→ Para el conjunto $\{2,4\}$:

$g(3, \{2,4\}) \Rightarrow Min\,[L_{32} + g(2, \{4\}), L_{34} + g(4, \{2\})]$

$\hookrightarrow [4+13,\ 6+10] = [17, 16] = 16$

→ Para el conjunto $\{3,4\}$:

$g(2, \{3,4\}) \Rightarrow Min\,[L_{23} + g(3, \{4\}), L_{24} + g(4, \{3\})]$

$\hookrightarrow [4+14,\ 5+14] = [18, 19] = 18$

Por último, ya podemos calcular la distancia mínima que pase por 3 nodos y el origen:

$$g(1, \{2,3,4\}) = Min \begin{bmatrix} L_{12} + g(2, \{3,4\}) \\ L_{13} + g(3, \{2,4\}) \\ L_{14} + g(4, \{2,3\}) \end{bmatrix}$$

$\hookrightarrow [5+18,\ 1+16,\ 8+15] = [23, \underline{\underline{17}}, 22]$

⇒ El recorrido mínimo es 17

⇒ Con el camino $\{1, 3, 4, 2, 1\}$

David Martínez Díaz 44669191 J

- Ejercicio 3:

La programación dinámica se basa en definir su solución en términos de las soluciones de subproblemas del mismo (utilizado sobretodo para problemas n-etápicos).

- Ventajas:

⇒ Por tanto la PD, tiene como ventaja que se progresa etapa por etapa con sub-problemas que se diferencian entre si en una unidad en sus tamaños.

⇒ Además como en cada etapa se comparan los resultados obtenidos, con precedentes, siempre se obtiene la solución óptima.

⇒ Por otro lado, los problemas están encajados, no se repiten cálculos.

⇒ Por último, si utilizamos la propiedad del Principio del Óptimo de Bellman reduce la cantidad de cálculos que hay que hacer.

- Desventajas:

⇒ Una desventaja a destacar es que consume un gran uso de recursos, es decir, consume mucha memoria.

⇒ Se generan muchas sub-sucesiones de decisiones, lo que no es muy eficiente.

⇒ Por ultimo, mencionar una ventaja o desventaja que la eficiencia de los problemas de programación dinámica es polinómica