2.2.1 Elementos de programación dinámica

La definición correcta de *estado* permite considerar por separado cada estado, y garantiza que la solución sea factible para todos los estados. La definición de *estado* conduce al siguiente marco unificador de la programación dinámica.

Principio de optimalidad.

Las decisiones futuras para las etapas restantes formarán una política óptima independientemente de las políticas adoptadas en las etapas anteriores.

La implementación del principio de optimalidad es evidente en los cálculos del ejemplo 2.2.1 de la sección anterior. En él subproblema 3, los cálculos recursivos del nodo 7 utilizan la distancia más corta a los nodos 5 y 6, es decir, los estados del subproblema 2, sin preocuparse como se llega a ellos desde el nodo 1.

Este principio no toma los detalles de cómo se optimiza un subproblema, en si la naturaleza de este es genérica y puede ser lineal o no lineal, así como las alternativas de solución puede o no ser finita o infinita, el fin de este principio es descomponer el problema original en subproblemas más manejables computacionalmente hablando.

Recursividad Hacia Adelante (AVANCE) y Recursividad hacia atrás (RETROCESO)

En el ejemplo 2.2.1 se empleó recursividad hacia adelante para realizar los cálculos de cada subproblema partiendo desde el 1 hasta el 3, sin embargo, este también puede resolverse por medio de la recursividad hacia atrás, comenzando desde subproblema tres y terminando en el 1.

La recursividad hacia adelante y hacia atrás deberán de dar una solución óptima, la recursividad hacia adelante parece más lógica para resolver problemas, sin embargo, en mucha de la literatura de la PD acerca del tema se emplea más la recursividad hacia atrás, la razón, pues computacionalmente parece ser más eficiente.

Solución recursiva hacia atrás del problema 2.2.1

La ecuación recursiva inversa es

$$f_4(x_4 = 7) = 0$$

$$f_i(x_i) = \min_{\substack{todas\ factibles \\ rutas\ (x_i, x_{i+1})}} \{d(x_i, x_{i+1}) + f_{i+1}(x_{i+1})\}, i = 1,2,3$$

El orden de los cálculos e $f_3 \longrightarrow f_2 \longrightarrow f_1$

Subproblema 3. Para el nodo 7 $(x_4 = 7)$ que está conectado con los nodos 5 y 6 $(x_3 = 5\ y\ 6)$, cada uno con una ruta cuyos cálculos se muestran en la tabla

	$d(x_3, x_4)$	Solución óptima	
x_3	$x_4 = 7$	$f_3(x_3)$	x_4^*
5	9	9	7
6	6	6	7

Subproblema 2. Se puede ver que la ruta entre los nodos 2 y 6 no existe, dado $f_3(x_3)$ desde el subproblema 3 se obtiene la tabla

	$d(x_2, x_3) + f_3(x_3)$		Solución óptima	
x_2	$x_3 = 5$	$x_3 = 6$	$f_2(x_2)$	x_3^*
2	$12 + 9 = 21$	—	21	5
3	$8 + 9 = 17$	$9 + 6 = 15$	15	6
4	$7 + 9 = 16$	$13 + 6 = 19$	16	5

Para la obtener la solución en las ciudades 2 y 4, la ruta más corta pasa por la ciudad 5 y para la ciudad 3 la ruta más corta pasa por la ciudad 6.

Subproblema 1. Partiendo del nodo 1, se tienen las rutas alternativas (1, 2), (1, 3) y (1, 4), empleando $f_2(x_2)$ del subproblema 2, se obtiene la tabla

	$d(x_1, x_2) + f_2(x_2)$			Solución óptima	
x_1	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$f_1(x_1)$	x_2^*
1	$7 + 21 = 28$	$8 + 15 = 23$	$5 + 16 = 21$	21	4

La solución del subproblema 1 conecta la ciudad 1 con la ciudad 4. La solución del subproblema 2 conecta la ciudad 4 con la ciudad 5 y por último, la solución del subproblema 3 conecta la ciudad 5 con la ciudad 7, por lo tanto la ruta optima del problema en RETROCESO es $1 \longrightarrow 4 \longrightarrow 5 \longrightarrow 7$ con una distancia total de 21 millas.

De acuerdo a lo visto hasta el momento hay tres elementos básicos para la programación dinámica

1. Definición de subproblemas
2. Definición de alternativas en cada subproblema
3. Definición de estados en cada subproblema

De estos tres elementos, la definición de estado es la más sutil y es la que debe ser modelada, de tal forma que deberá preguntarse:

¿Qué relaciones ligan a los subproblemas entre sí?

¿Qué información se necesita para tomar decisiones factibles en el subproblema actual independiente de las tomadas en los subproblemas anteriores?

Para mejorar la comprensión del estado, cuestione la validez del mismo, pruebe la definición más lógica y utilícela en los cálculos recursivos, pues que el rol de los estados es el más importante para dar solución a un problema.

TAREA.

Resolver y programar (en el mismo código anterior) la tarea de las carreteras del punto anterior empleando RETROCESO, solo incluya la opción al usuario de como desea resolverlo, si por AVANCE o RETROCESO