TPN°6: Programación Dinámica

Algoritmos y Estructuras de Datos

PROGRAMACIÓN DINÁMICA

Estrategia para resolver **problemas de optimización**, que **divide en subproblemas** más pequeños, los cuáles resuelve, **guardando los resultados** para no calcularlos más de una vez

PROGRAMACIÓN DINÁMICA

PROGRAMACIÓN DINÁMICA

PRINCIPIO DE OPTIMALIDAD

En una secuencia de decisiones óptima toda subsecuencia ha de ser también óptima

- Puede aplicarse a problemas donde falla D&C y Greedy
  - Gran cantidad de subproblemas
  - Subproblemas cuyas soluciones parciales se solapan
  - Grupos de subproblemas de distinta complejidad
  - Se producen varias secuencias de decisiones y al final se sabe cuál es la mejor de ellas
- Eficiente
- Resuelve el problema etapa por etapa en forma sistemática

PROGRAMACIÓN DINÁMICA

EJEMPLOS DE APLICACIÓN

- Fibonacci
- Dar vuelto a un cliente
- Cálculo de coeficientes binomiales
- Algoritmo de Floyd
- Algoritmo de Warshall
- Problema de la mochila múltiple
- Problema de la mochila 0/1
- Multiplicación óptima de matrices

PROGRAMACIÓN DINÁMICA

Dado un monto X en pesos, indicar cuál es la menor cantidad de monedas

que son necesarias para pagarlo exactamente

(asumiendo que tengo **cantidad ilimitada** de cada tipo)

El algoritmo Greedy funciona solamente en número limitado de casos

PROGRAMACIÓN DINÁMICA

PROBLEMA DE MONEDA

OBJETIVO ► Utilizar la menor cantidad de monedas para formar un monto X

- Tengo que entregar un monto X en pesos
- Denominación de las monedas: D = {$2, $3, $5}
- Cantidad **ilimitada** de monedas

X =	0	1	2	…	Vuelto
d_1					
d_2					
…				**C(i,j)**	
d_n					

PROGRAMACIÓN DINÁMICA

$$C(i, j) = \begin{cases} \infty & , j<0 \ \vee \ i \leq 0 \\ 0 & , j=0 \\ \min\left(C(i-1, j), 1 + C(i, j-di) \right) & \end{cases}$$

……. ¿Qué sucede con la primera fila de la tabla?

X =	0	1	2	3	4	5	6	7
$d_1 = 2$	0	∞	**1**	∞	**2**	∞	**3**	∞
$d_2 = 3$	0	∞	**1**	**1**	**2**	**2**	**2**	**3**
$d_3 = 5$	0	∞	**1**	**1**	**2**	**1**	**2**	**2**

PROGRAMACIÓN DINÁMICA

¿Cómo obtengo las monedas que tengo que entregar como vuelto?

X =	0	1	2	3	4	5	6	7
$d_1 = 2$	0	∞	**1**	∞	**2**	∞	**3**	∞
$d_2 = 3$	0	∞	**1**	**1**	**2**	**2**	**2**	**3**
$d_3 = 5$	0	∞	**1**	**1**	**2**	**1**	**2**	**2**

?

?

?

S = {$5 , $2}

PROGRAMACIÓN DINÁMICA

Dado un monto X en pesos, indicar cuál es la menor cantidad de monedas

que son necesarias para pagarlo exactamente

Cantidad **limitada** de monedas de cada denominación

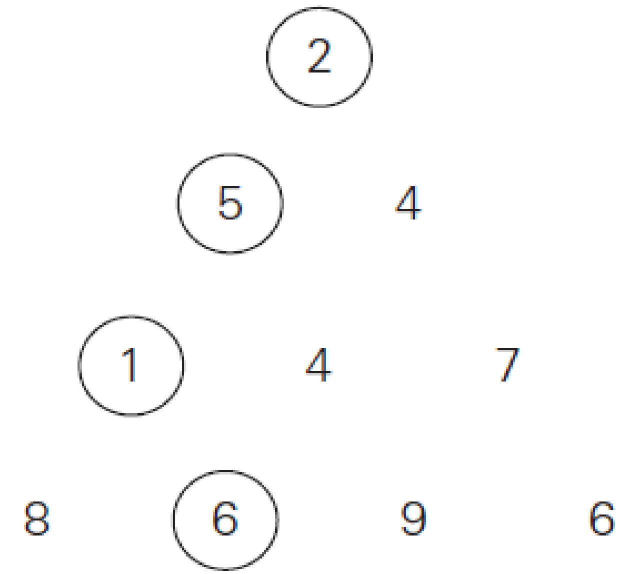
PROGRAMACIÓN DINÁMICA

$$C(i, j) = \begin{cases} \infty & , j<0 \ \vee \ i\leq 0 \\ 0 & , j=0 \\ \min\limits_{k = 0,..m_i} (\, k + C(i-1, j-k*di)\,) & \\ \text{……. ¿Qué sucede con la primera fila de la tabla?} & \end{cases}$$

X =	0	1	2	3	4	5	6	7
$d_1 = 2$ $m_1 = 2$	0	∞	**1**	∞	**2**	∞	∞	∞
$d_2 = 3$ $m_2 = 6$	0	∞	**1**	**1**	**2**	**2**	**2**	**3**
$d_3 = 5$ $m_3 = 3$	0	∞	**1**	**1**	**2**	**1**	**2**	**2**

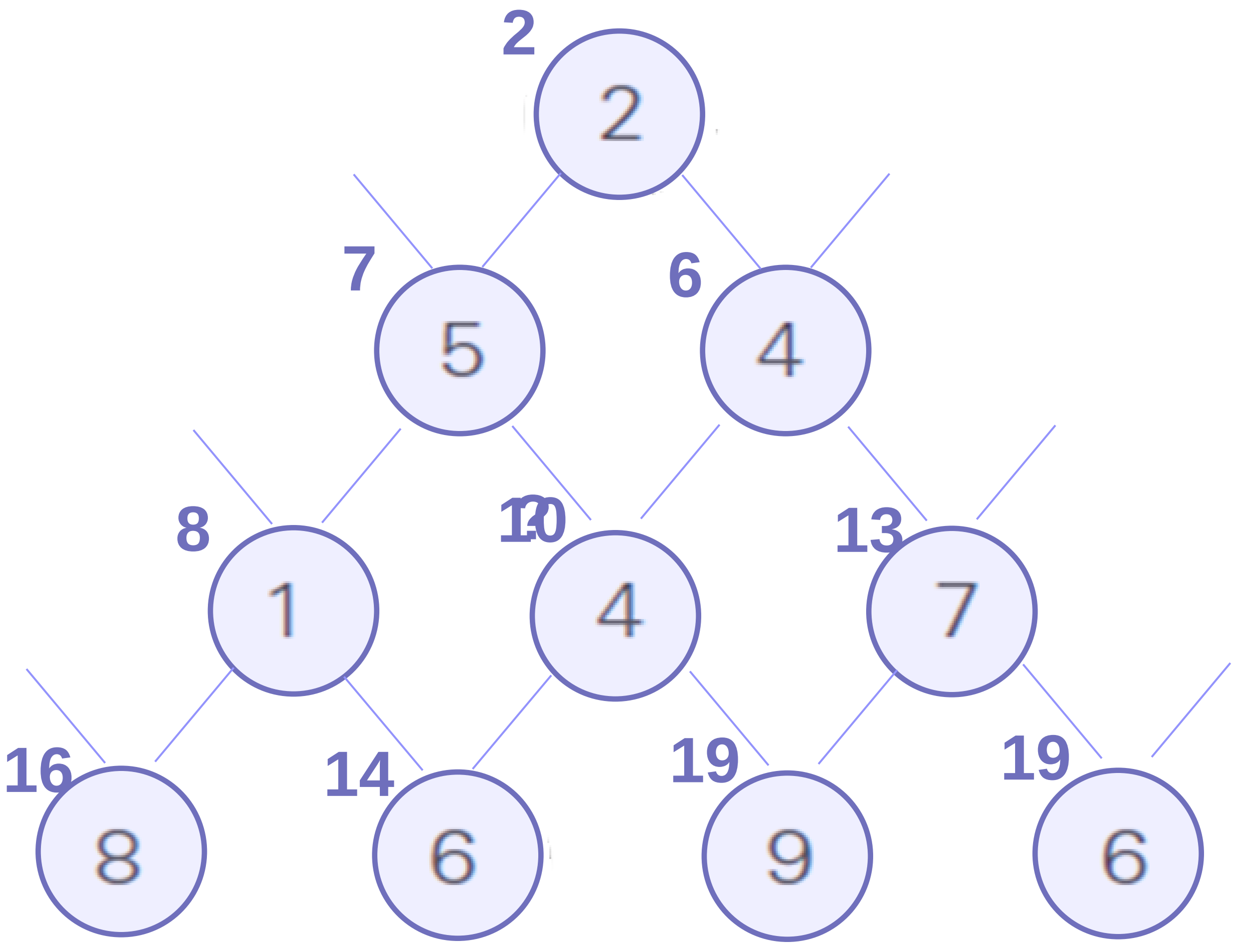
PROGRAMACIÓN DINÁMICA

Se disponen números enteros positivos en un triángulo equilátero de **base n**.



Se desea encontrar la **suma mínima** descendiendo desde el vértice superior del triángulo hasta su base a través de una secuencia de números adyacentes (indicados en la figura mediante círculos)

PROGRAMACIÓN DINÁMICA



PROGRAMACIÓN DINÁMICA

PROBLEMA SUMA MINIMA TRIANGULO

OBJETIVO ► Lograr la suma mínima desde el vértice superior del triángulo hasta su base a través de una secuencia de números adyacentes

Tengo que encontrar cuáles son los números que forman esa suma mínima

ALGORITMO

Entrada: Matriz triangular inferior Tnxn

Salida: suma mínima

PROGRAMACIÓN DINÁMICA

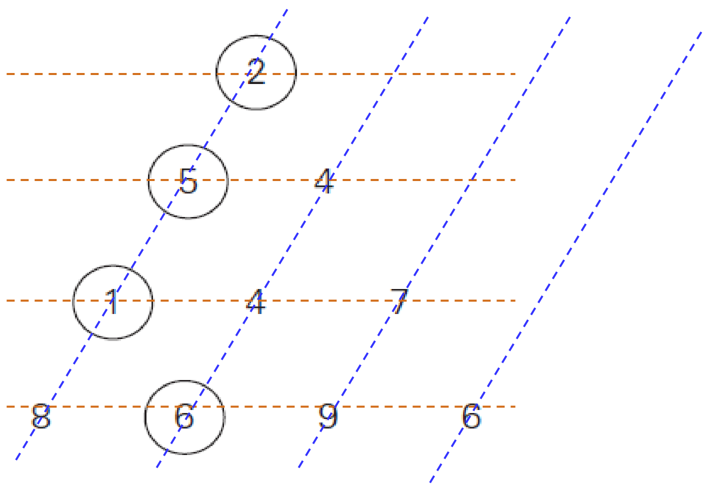


Tabla de
Suma Mínima

Entrada

T$_{4x4}$

Columnas

Filas (niveles)

2	0	0	0
5	4	0	0
1	4	7	0
8	6	9	6

Niveles

	0	1	2	3	4
1					
2					
3					
4					

PROGRAMACIÓN DINÁMICA

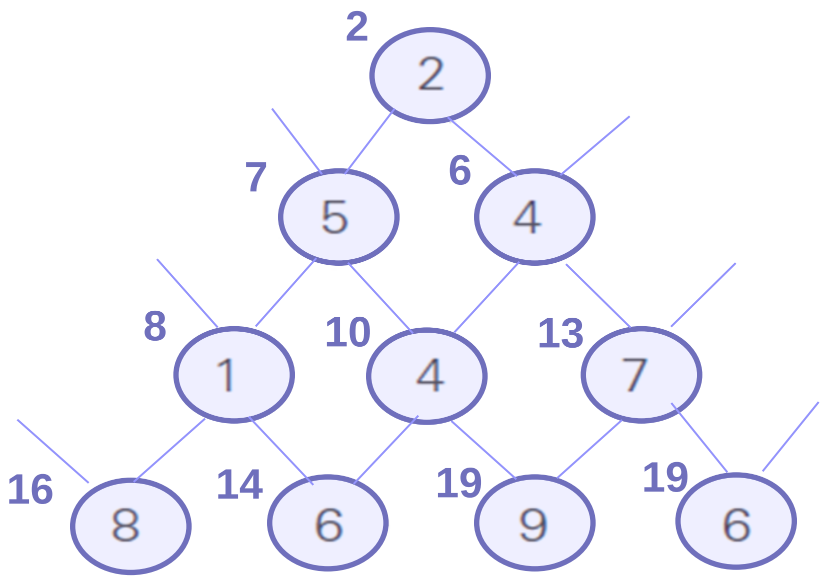


Tabla de Suma Mínima

Niveles

	0	1	2	3	4
1	∞	**2**	∞	∞	∞
2	∞	5+ **7**	4+ **6**	∞	∞
3	∞	**8**	**10**	**13**	∞
4	∞	**16**	**14**	**19**	**19**

Entrada

T_{4x4}

Columnas

Filas (niveles)

2	0	0	0
5	4	0	0
1	4	7	0
8	6	9	6

PROGRAMACIÓN DINÁMICA

$$C(i, j) = \begin{cases} \infty & , j=0 \vee j>i \\ T(i,j) & , i=1 \wedge j=1 \\ T(i,j) + \min(C(i-1, j-1), C(i-1, j)) & \end{cases}$$

Tabla de Suma Mínima

C

Niveles

	0	1	2	3	4
1	∞	**2**	∞	∞	∞
2	∞	**5+**		∞	∞
3	∞				∞
4	∞				

Preguntas…
…y a practicar…