



# PROBLEMA DE LA MOCHILA 0/1 (CON PROGRAMACIÓN DINÁMICA)

---

Algoritmos y Programación

Escuela de Ingeniería Informática

Universidad de Las Palmas de Gran Canaria

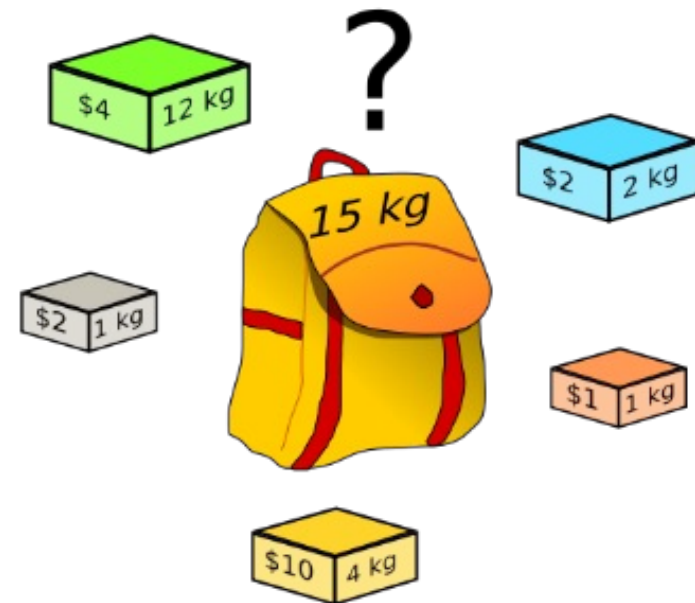
2 de noviembre de 2023

# Ejemplo

**Entrada:** Peso de N items  $\{w_1, w_2, \dots, w_n\}$   
Beneficio de N items  $\{b_1, b_2, \dots, b_n\}$   
Mochila con un límite W

**Salida:** Elección  $\{x_1, x_2, \dots, x_n\}$   
... donde  $x_i \in \{0, 1\}$ .

*Elección  
binaria*



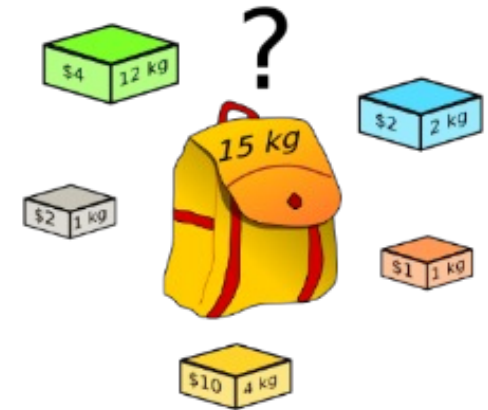
# Ejemplo

$W = 2, 3, 4, 5$

$B = 3, 4, 5, 6$

Peso

Beneficio



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*¿ Recurrencia ?*

W = 2, 3, 4, 5

B = 3, 4, 5, 6

N = 1, 2, 3, 4



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*

N=4

w=5

$W = 2, 3, 4, \underline{5}$

$B = 3, 4, 5, 6$

$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*

$N=4$   
 $w=5$

*Lo cojo*



$W = 2, 3, 4, \underline{5}$

$B = 3, 4, 5, 6$

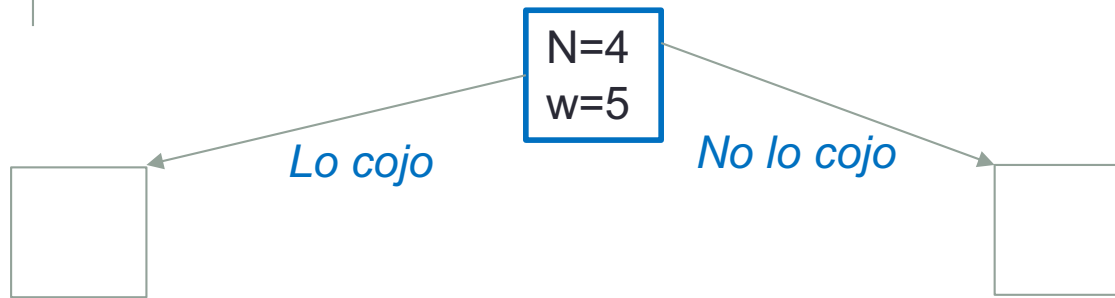
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, 3, 4, \underline{5}$

$B = 3, 4, 5, 6$

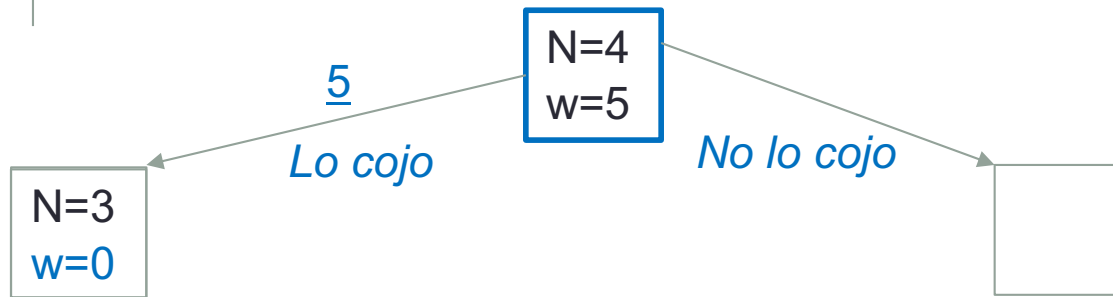
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, 3, 4, \underline{5}$

$B = 3, 4, 5, 6$

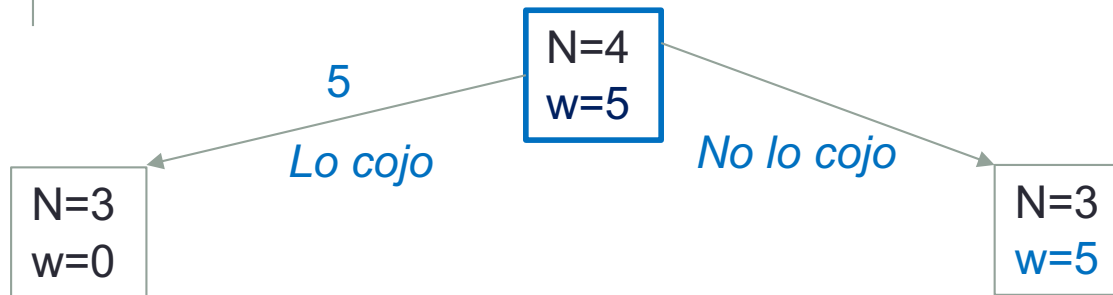
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*





$W = 2, 3, \underline{4}, 5$

Peso

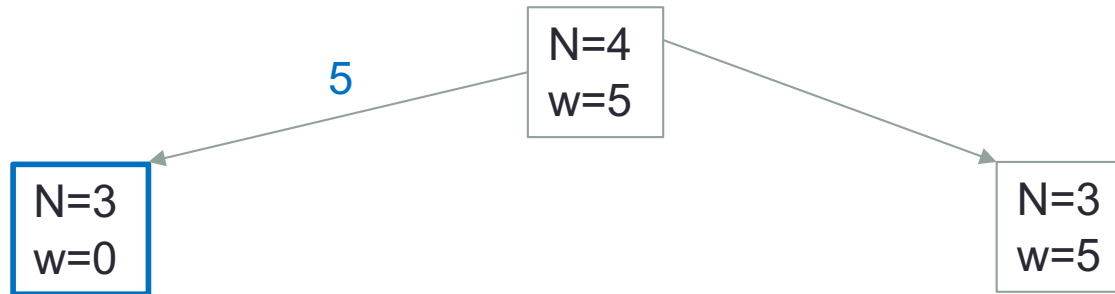
$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, 3, \underline{4}, 5$

$B = 3, 4, 5, 6$

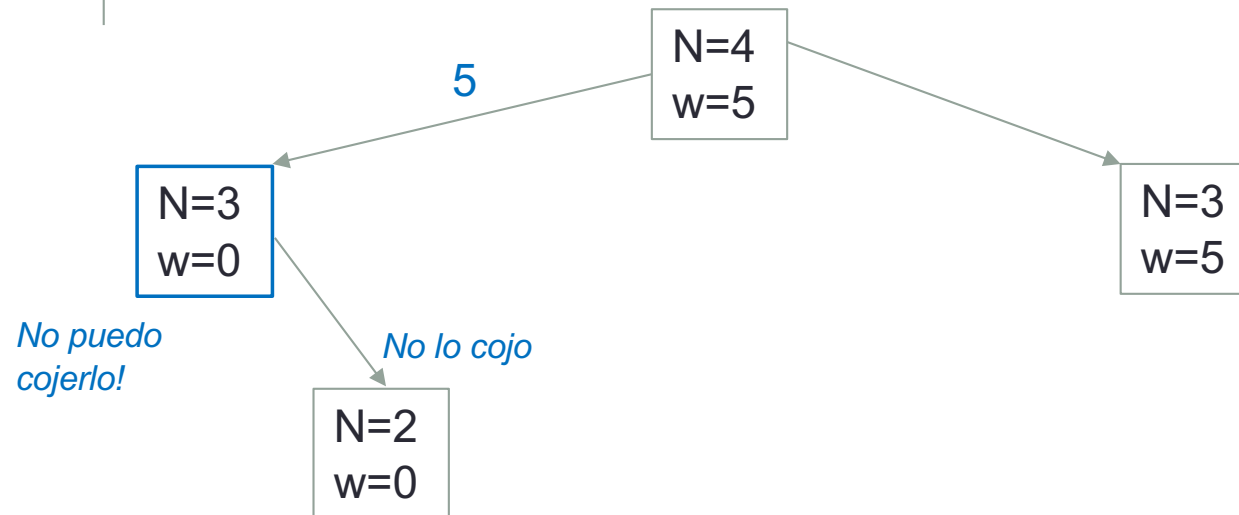
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, \underline{3}, 4, 5$

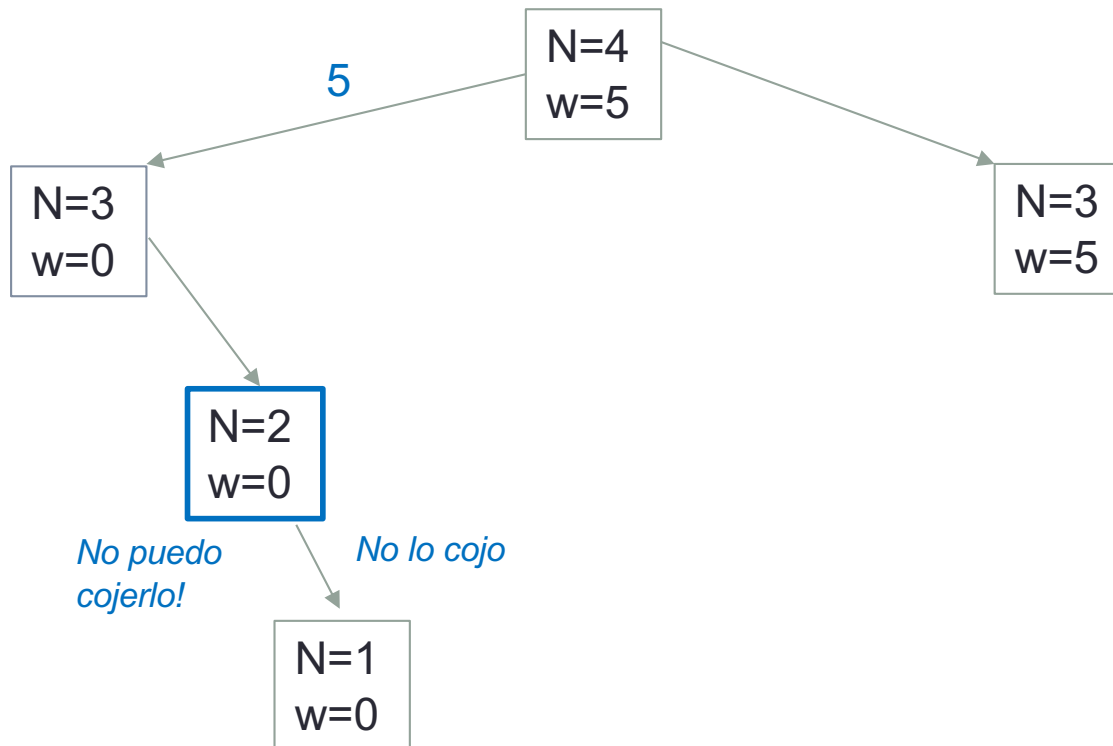
Peso

$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = \underline{2}, 3, 4, 5$

$B = 3, 4, 5, 6$

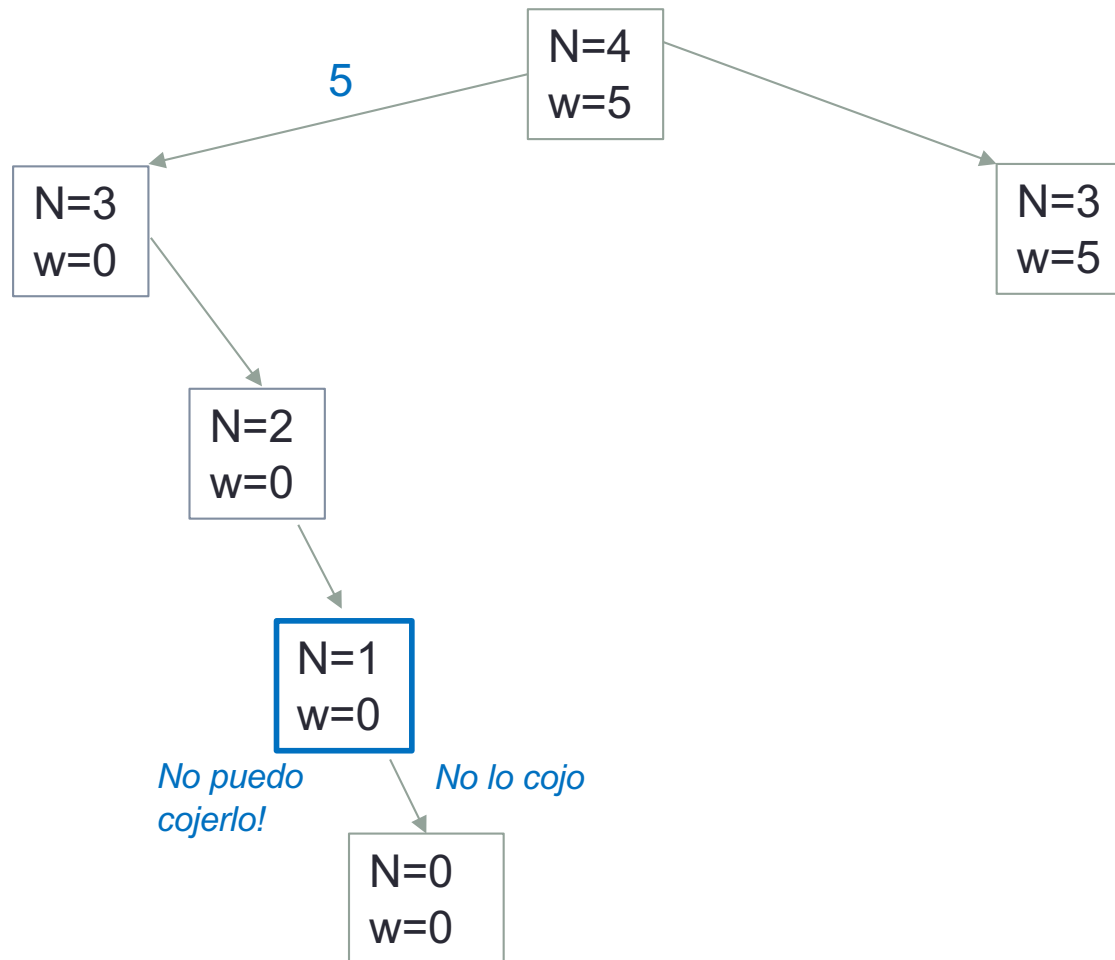
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, 3, 4, 5$

$B = 3, 4, 5, 6$

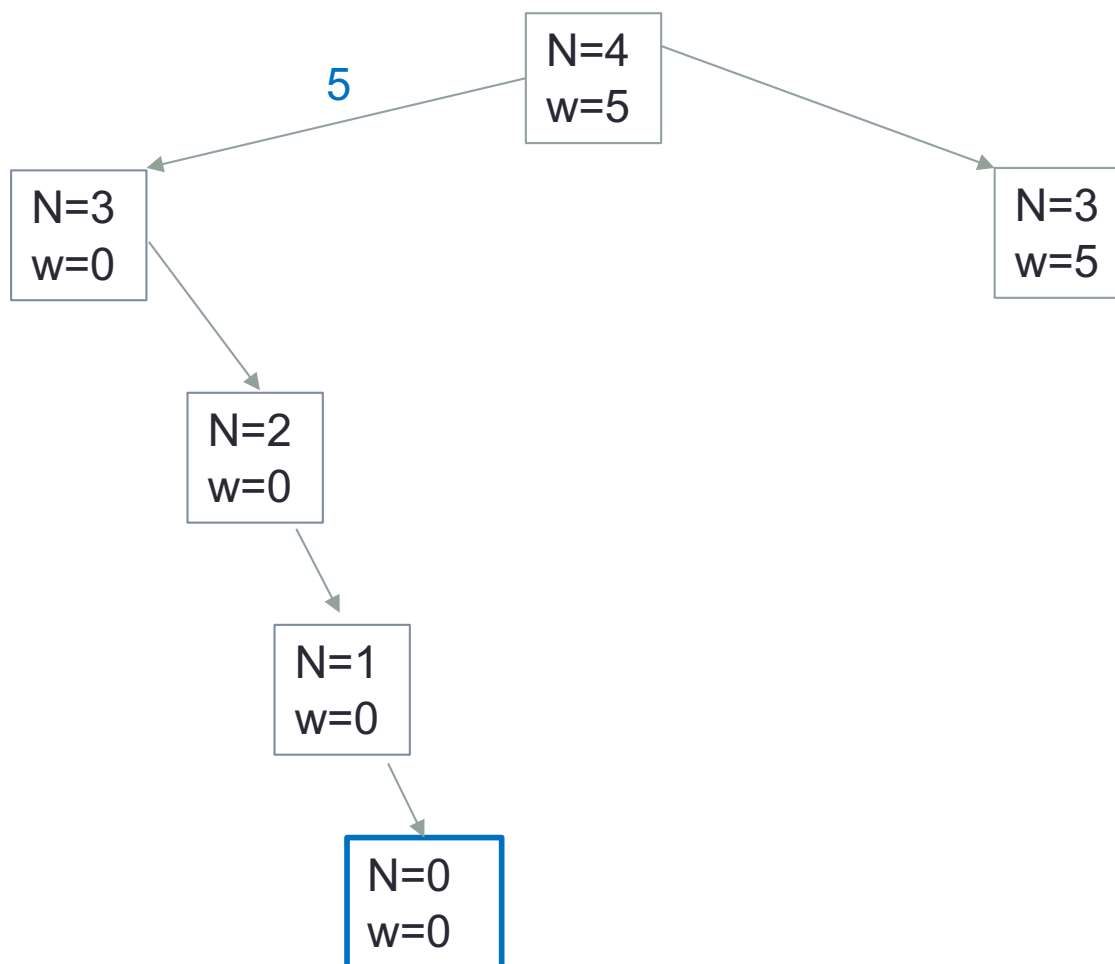
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$N = 0 \rightarrow$  Fin de los ítems

$W = 2, 3, \underline{4}, 5$

$B = 3, 4, 5, 6$

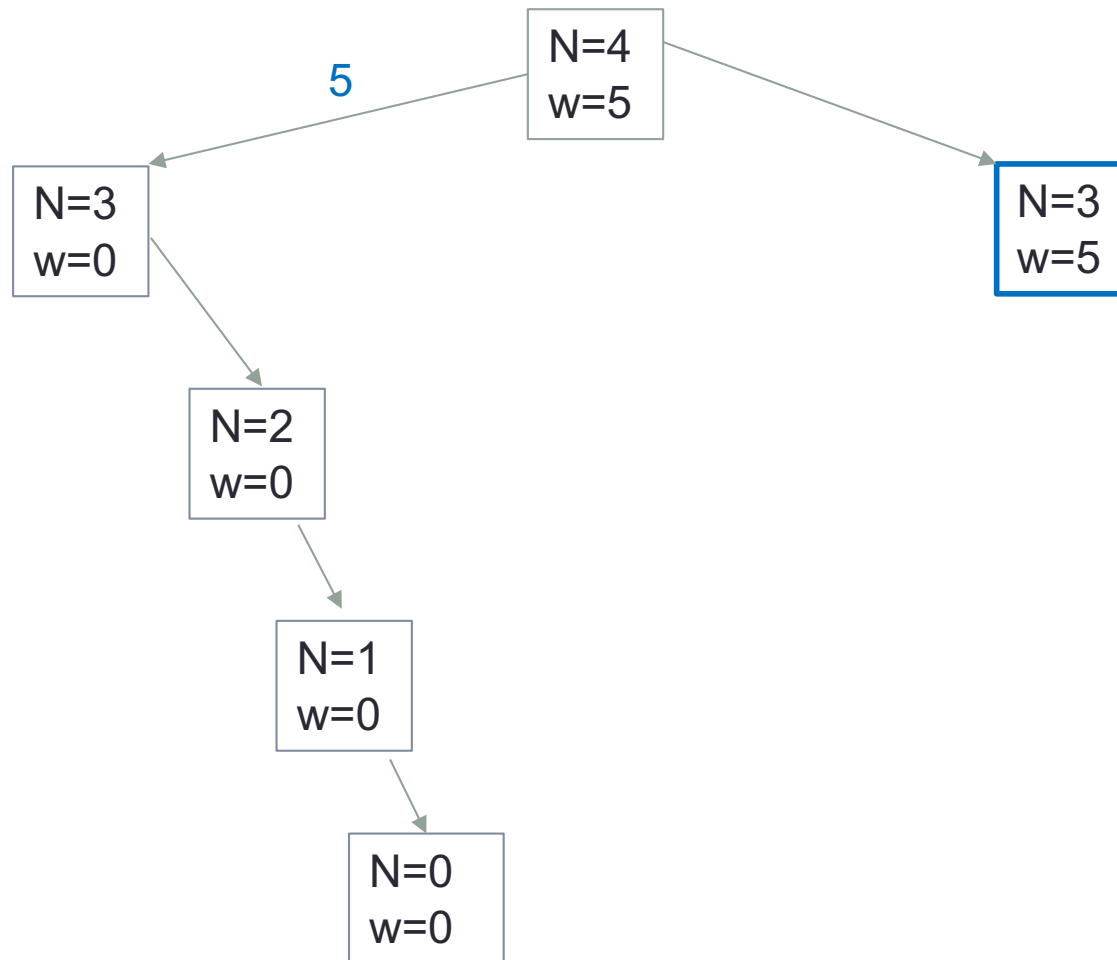
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, 3, \underline{4}, 5$

$B = 3, 4, 5, 6$

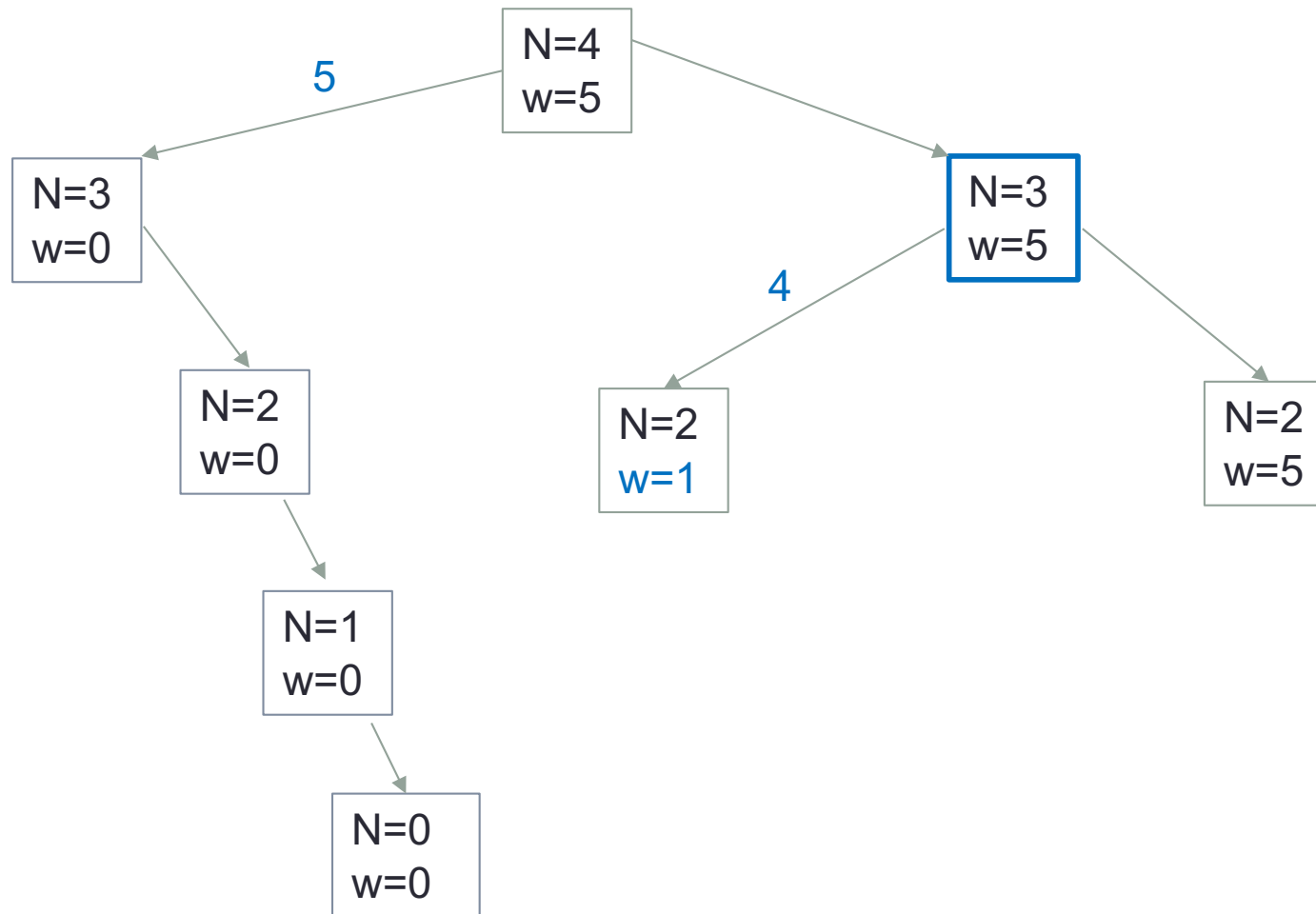
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, \underline{3}, 4, 5$

$B = 3, 4, 5, 6$

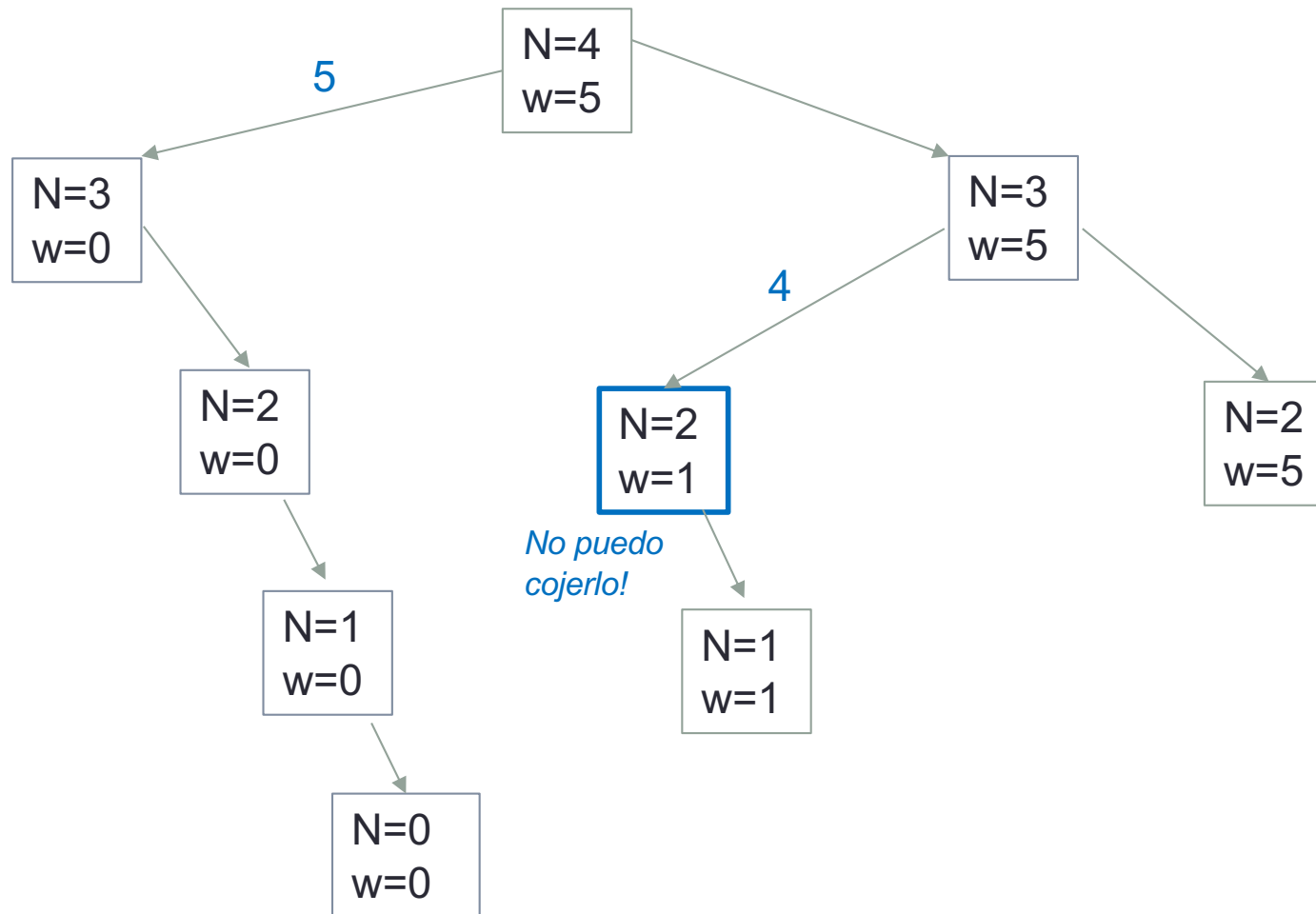
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*





$W = \underline{2}, 3, 4, 5$

Peso

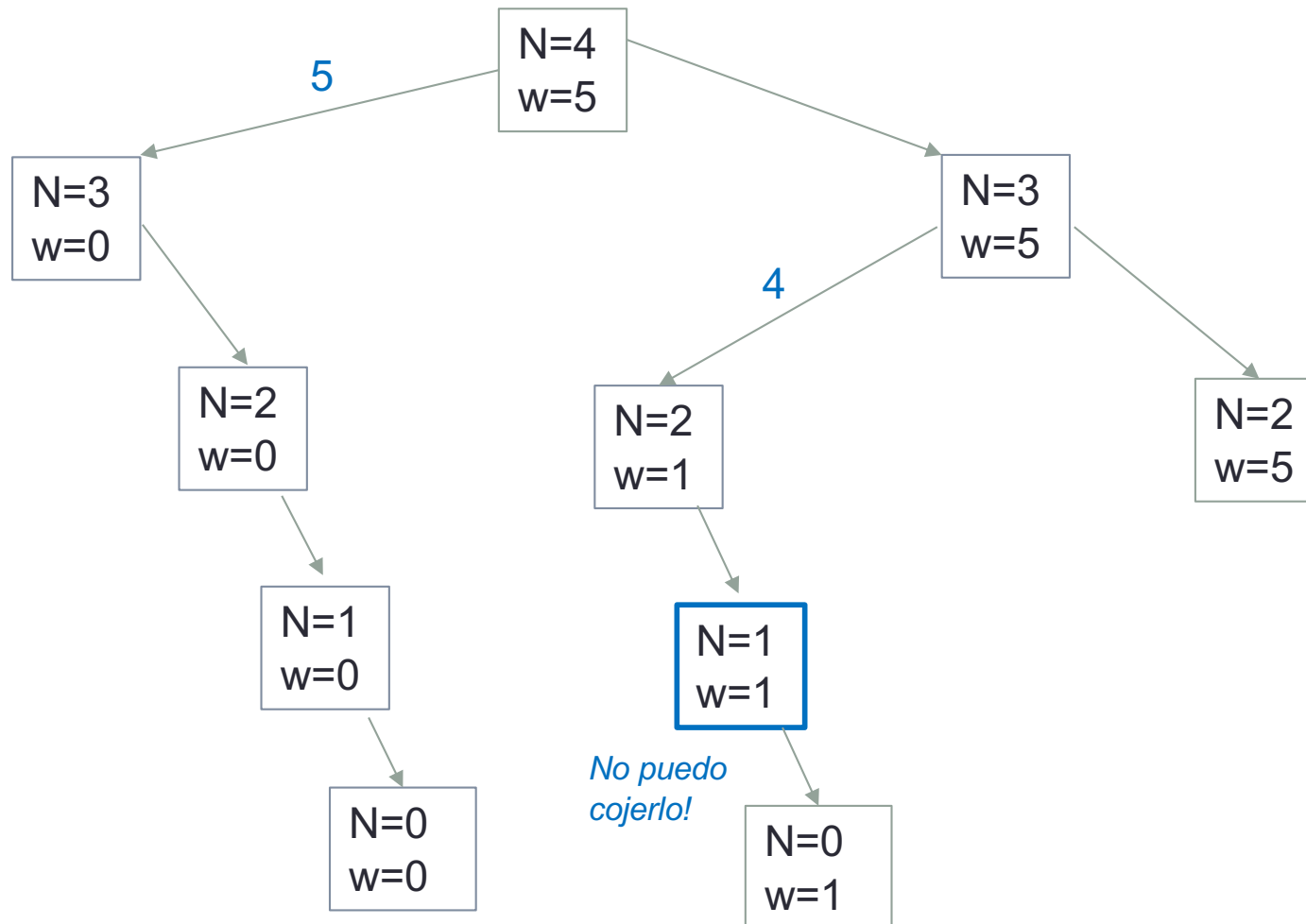
$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, 3, 4, 5$

Peso

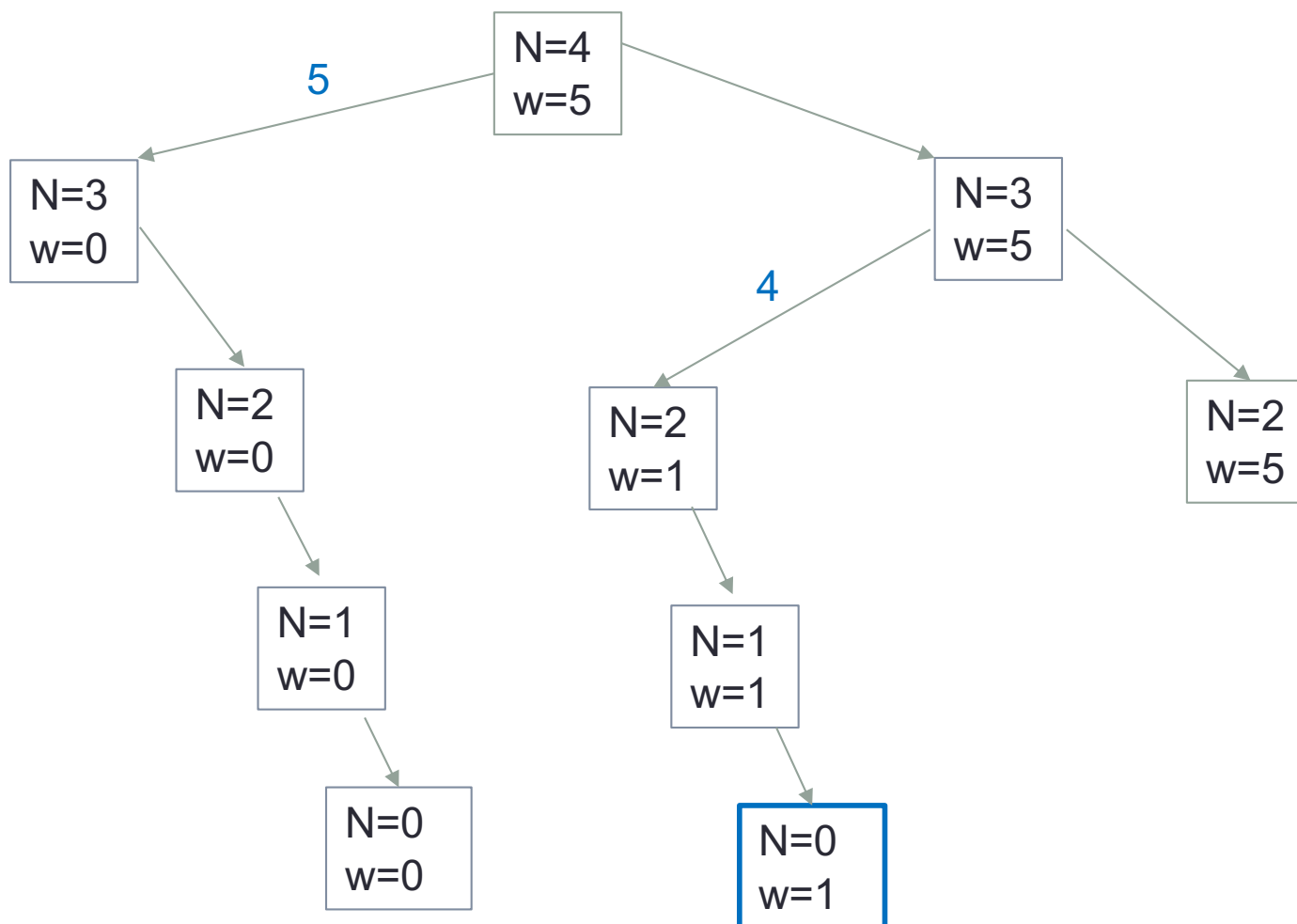
$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$N = 0 \rightarrow$  Fin de los items

$W = 2, \underline{3}, 4, 5$

$B = 3, 4, 5, 6$

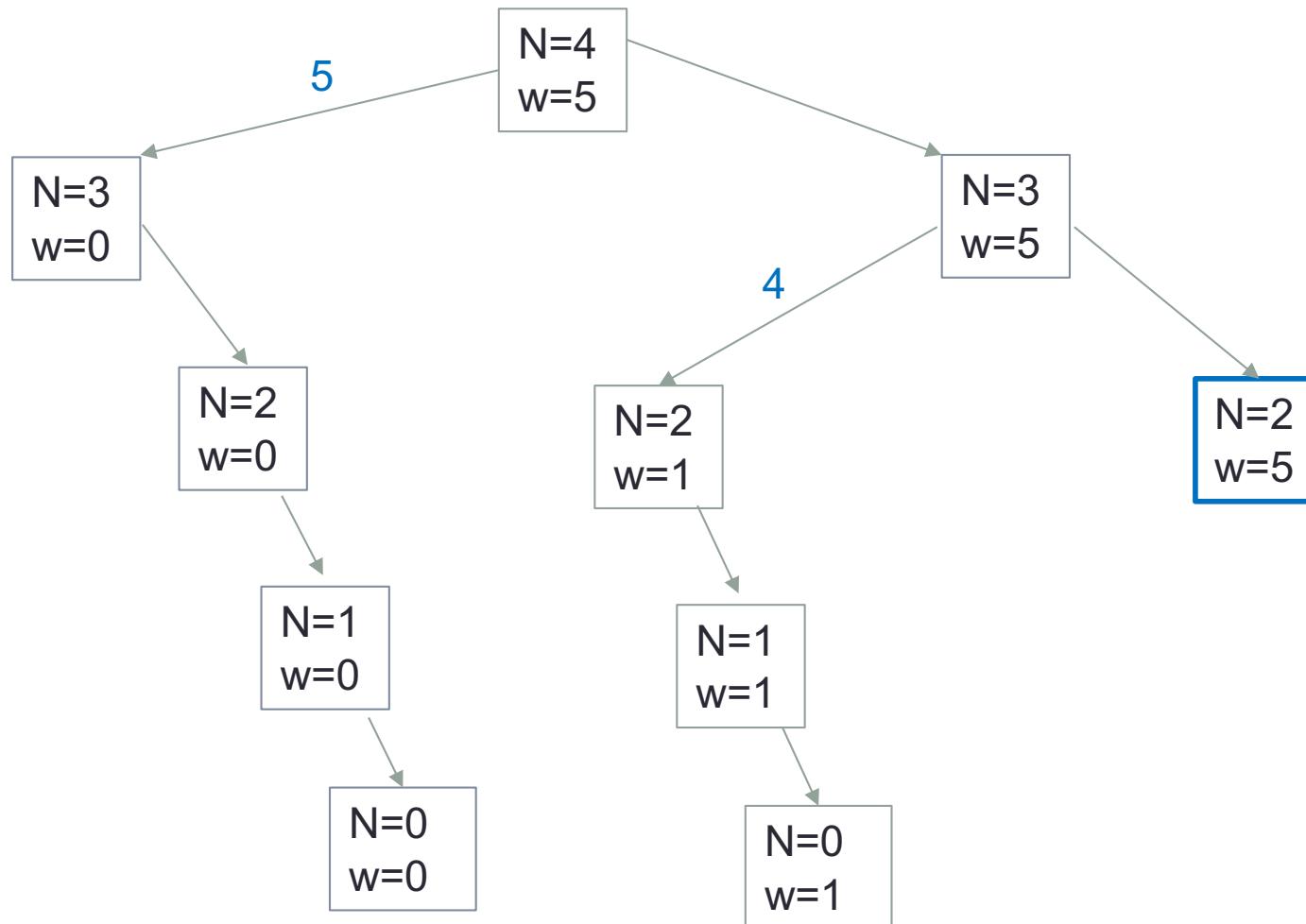
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, \underline{3}, 4, 5$

$B = 3, 4, 5, 6$

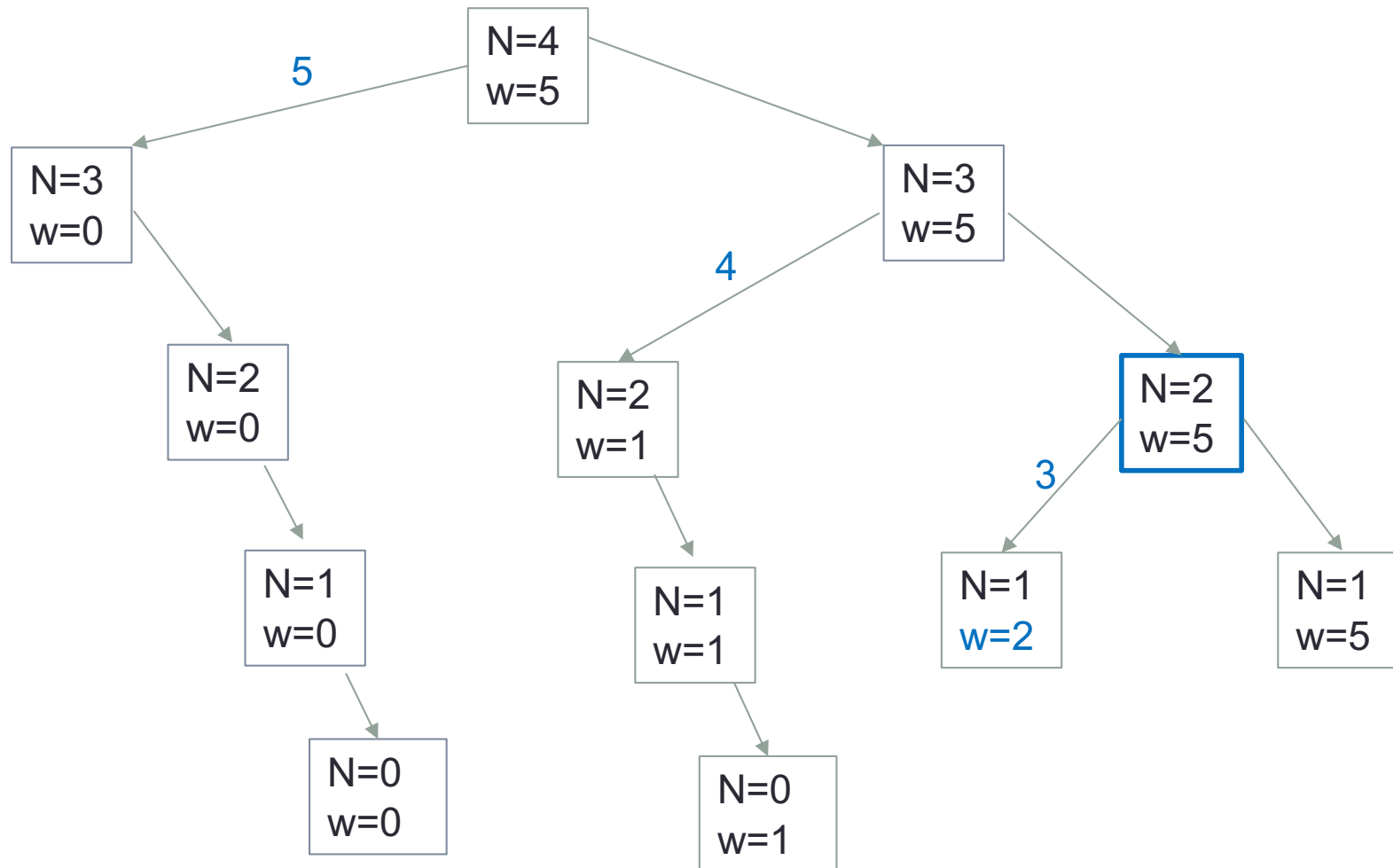
$N = 1, 2, 3, 4$



Peso

Beneficio

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = \underline{2}, 3, 4, 5$

Peso

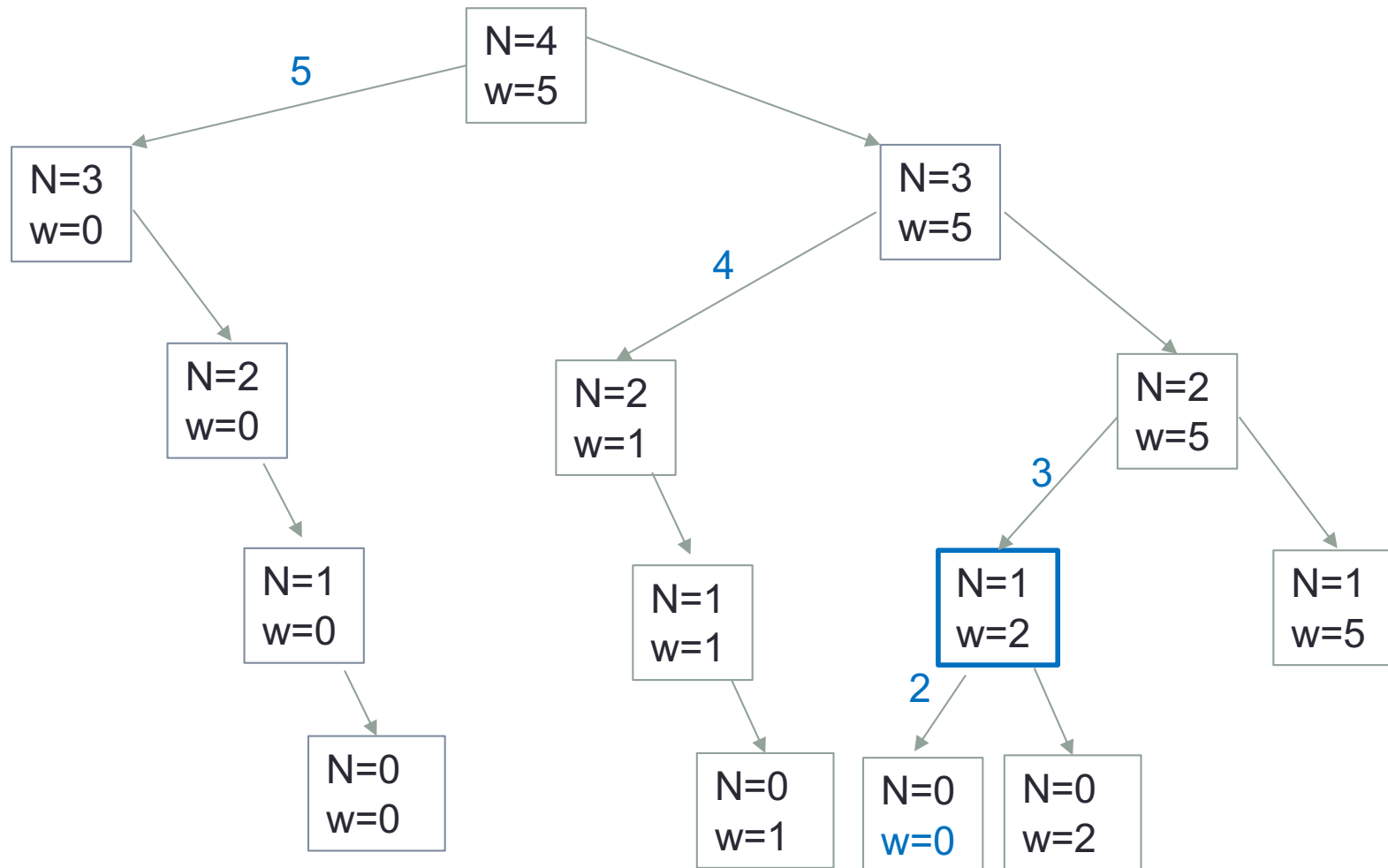
$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



$W = 2, 3, 4, 5$

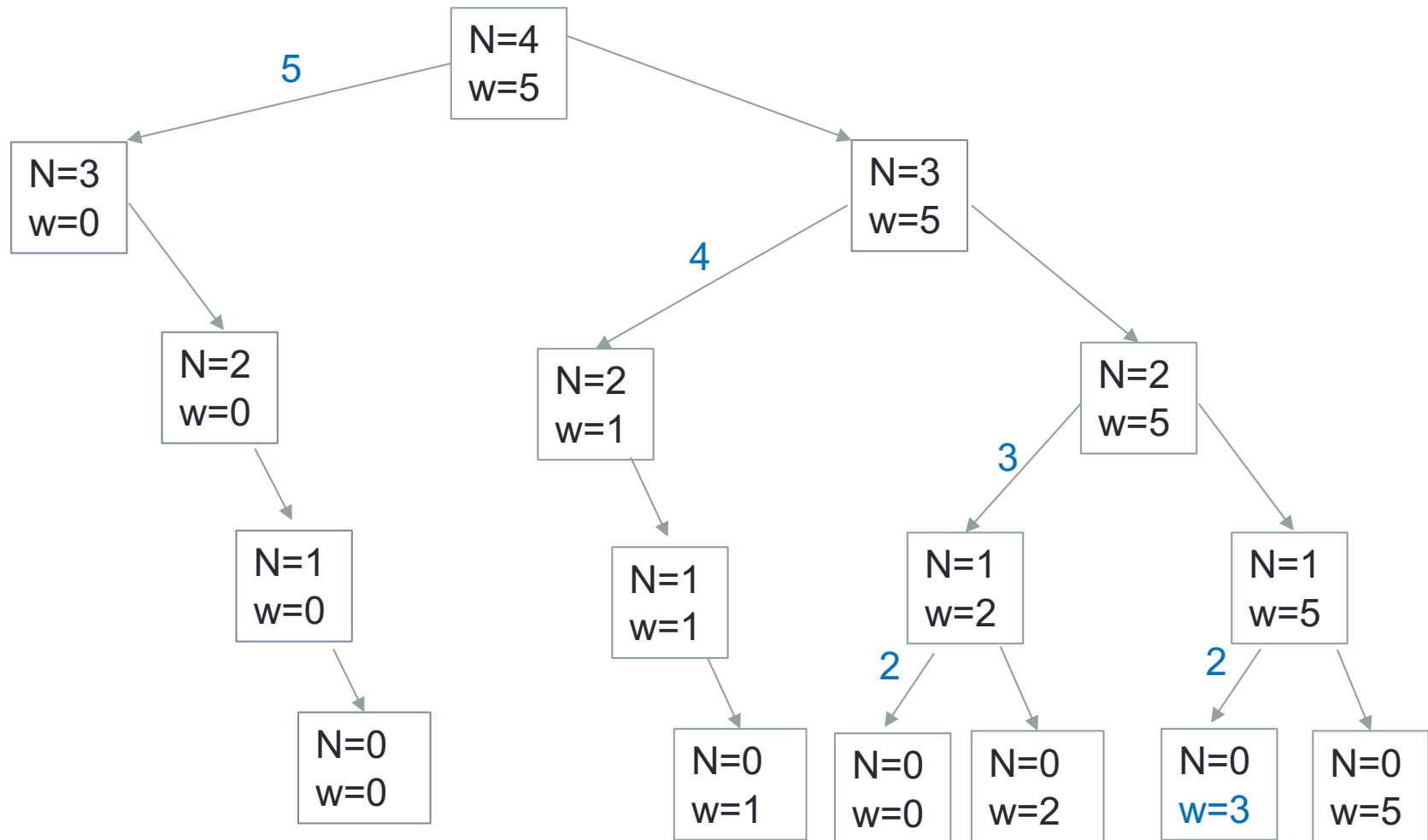
Peso

$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*¡ Árbol completo de llamadas !*

$W = 2, 3, 4, 5$

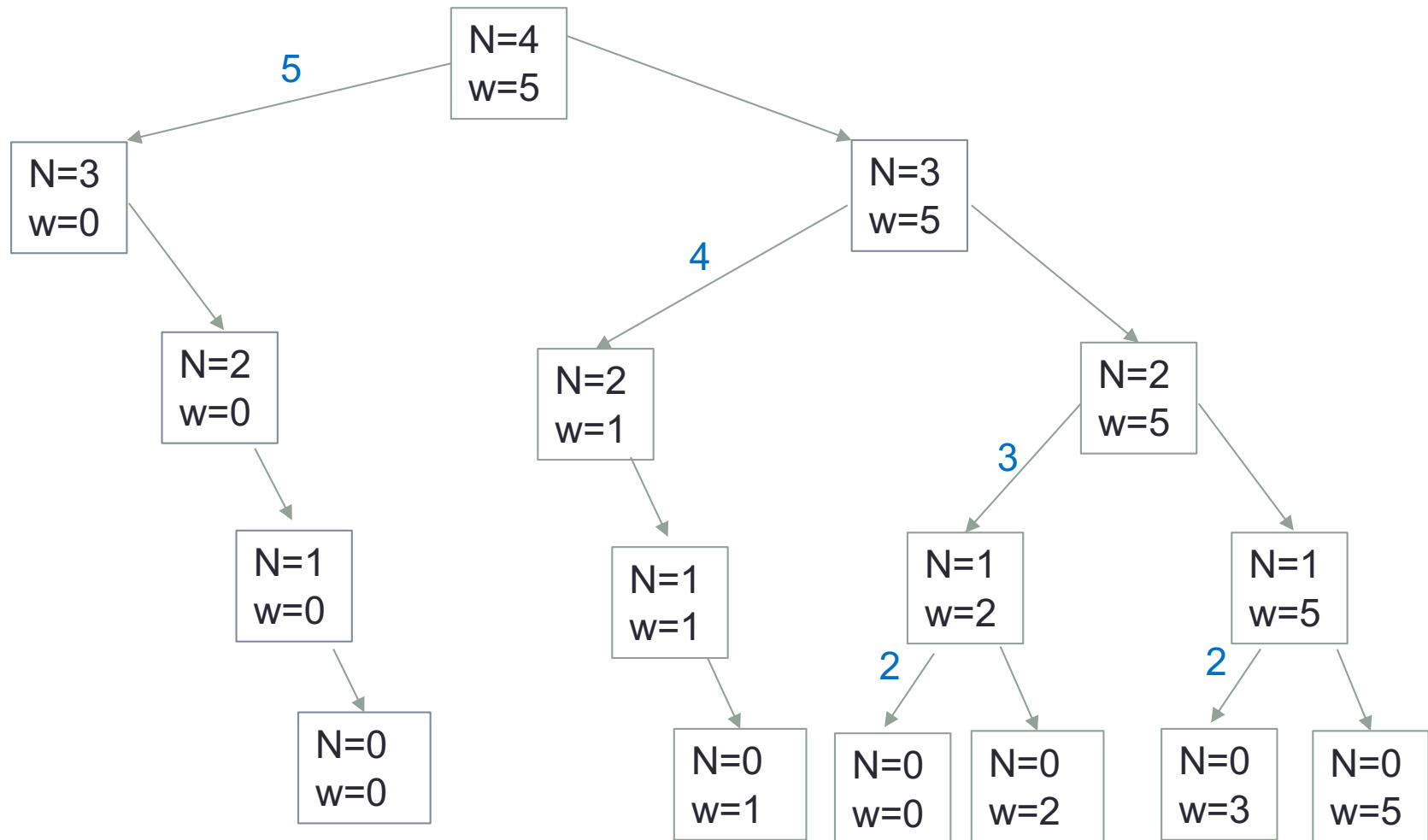
Peso

$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

$W = 2, 3, 4, 5$

Peso

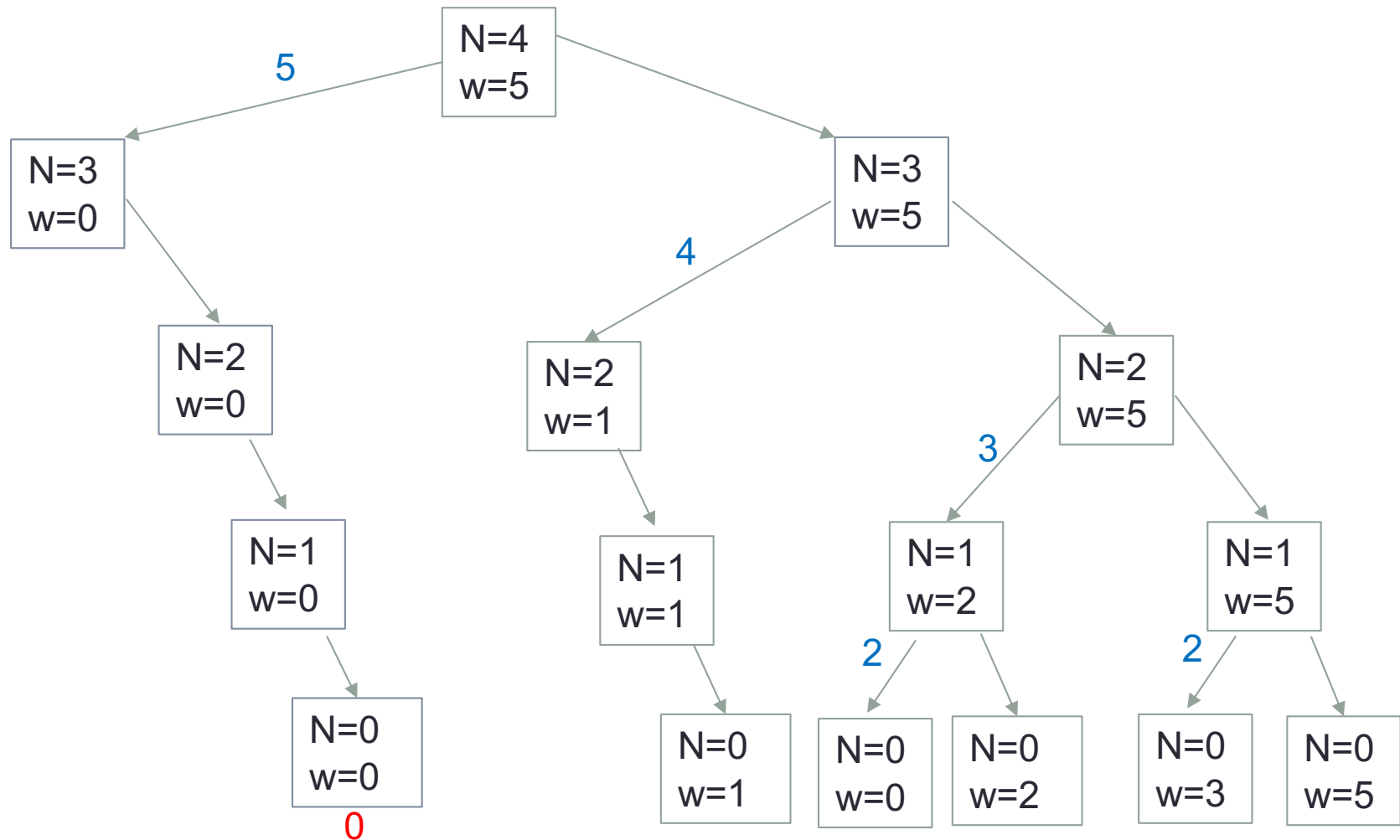
$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*



$W = 2, 3, 4, 5$

Peso

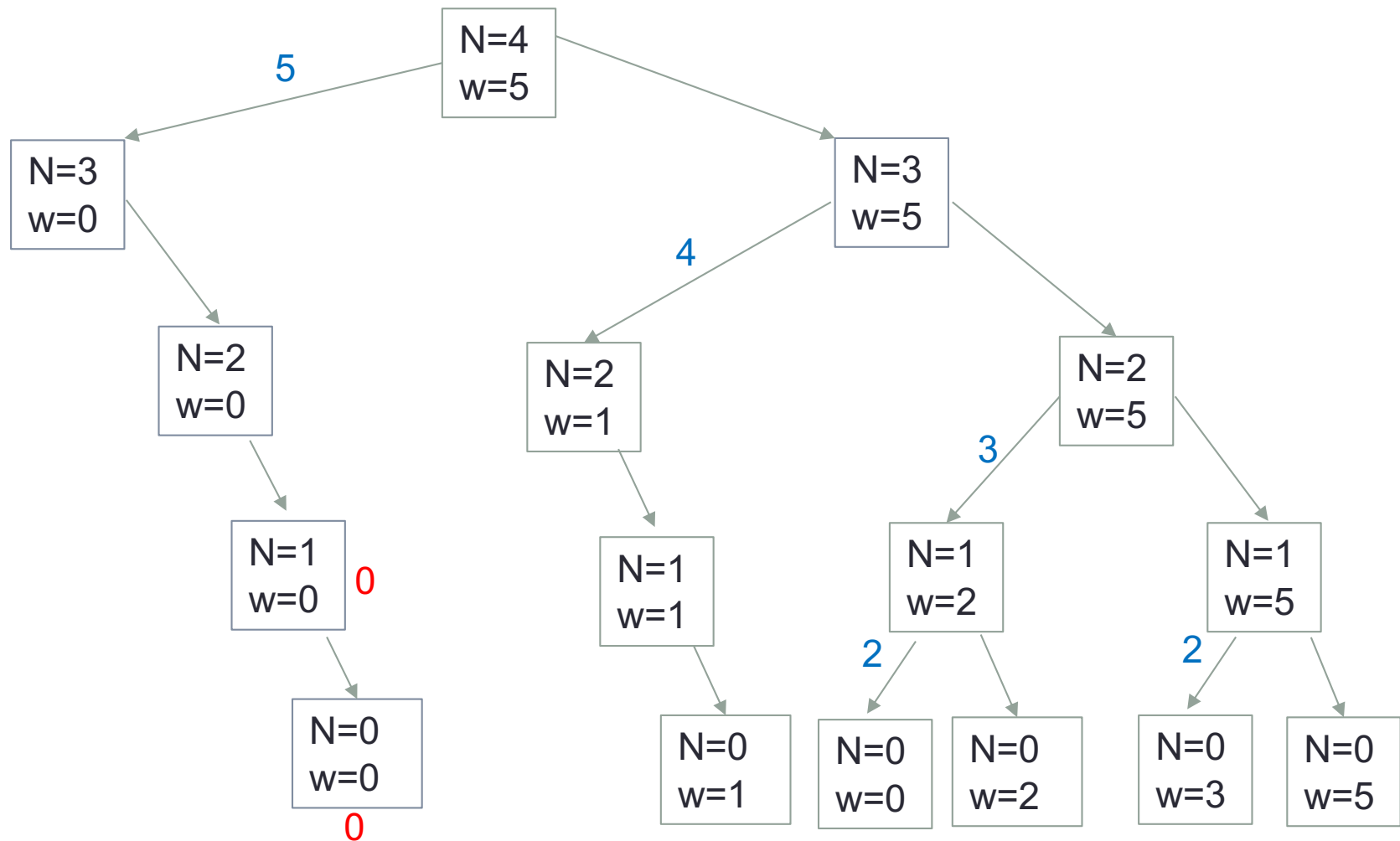
$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

$W = 2, 3, 4, 5$

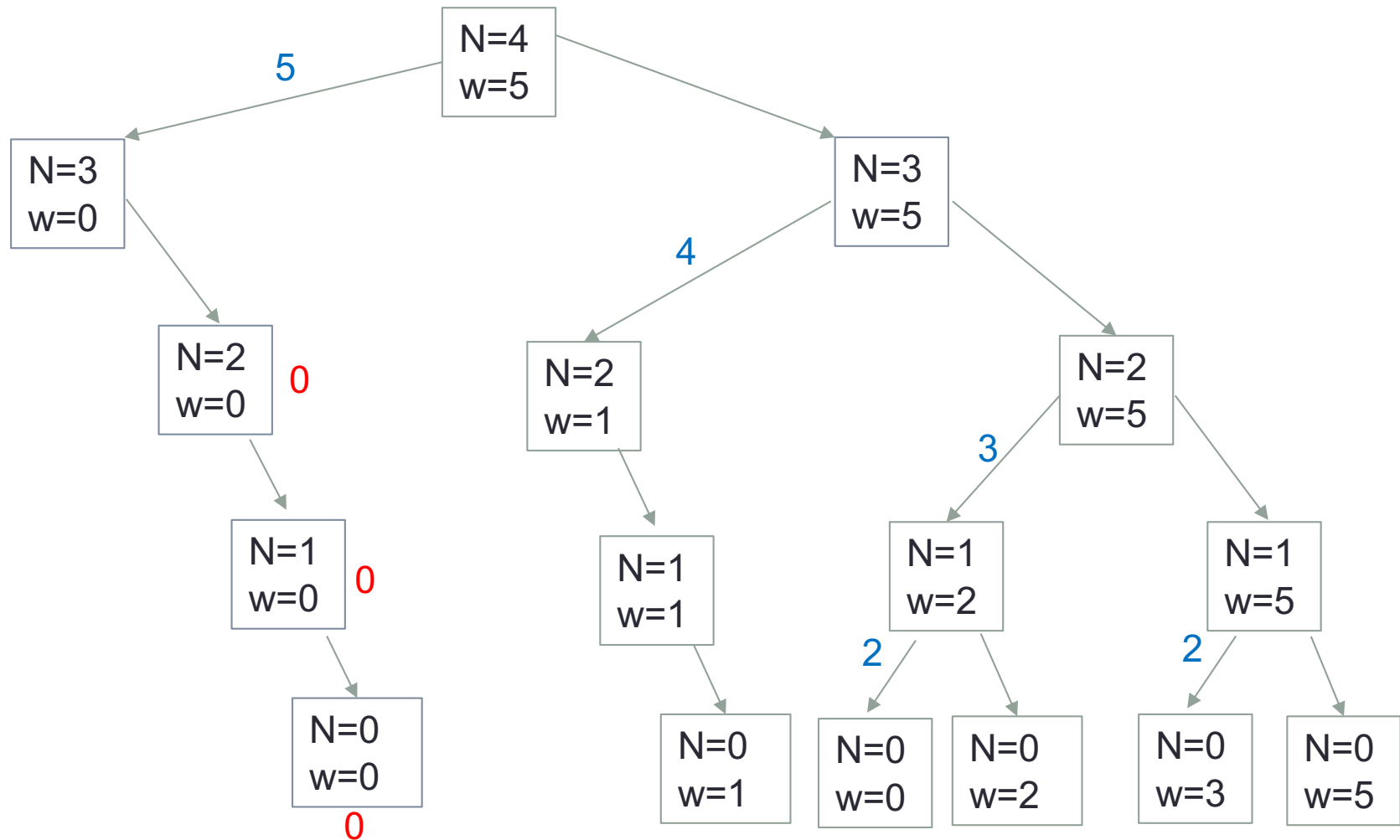
Peso

$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

$W = 2, 3, 4, 5$

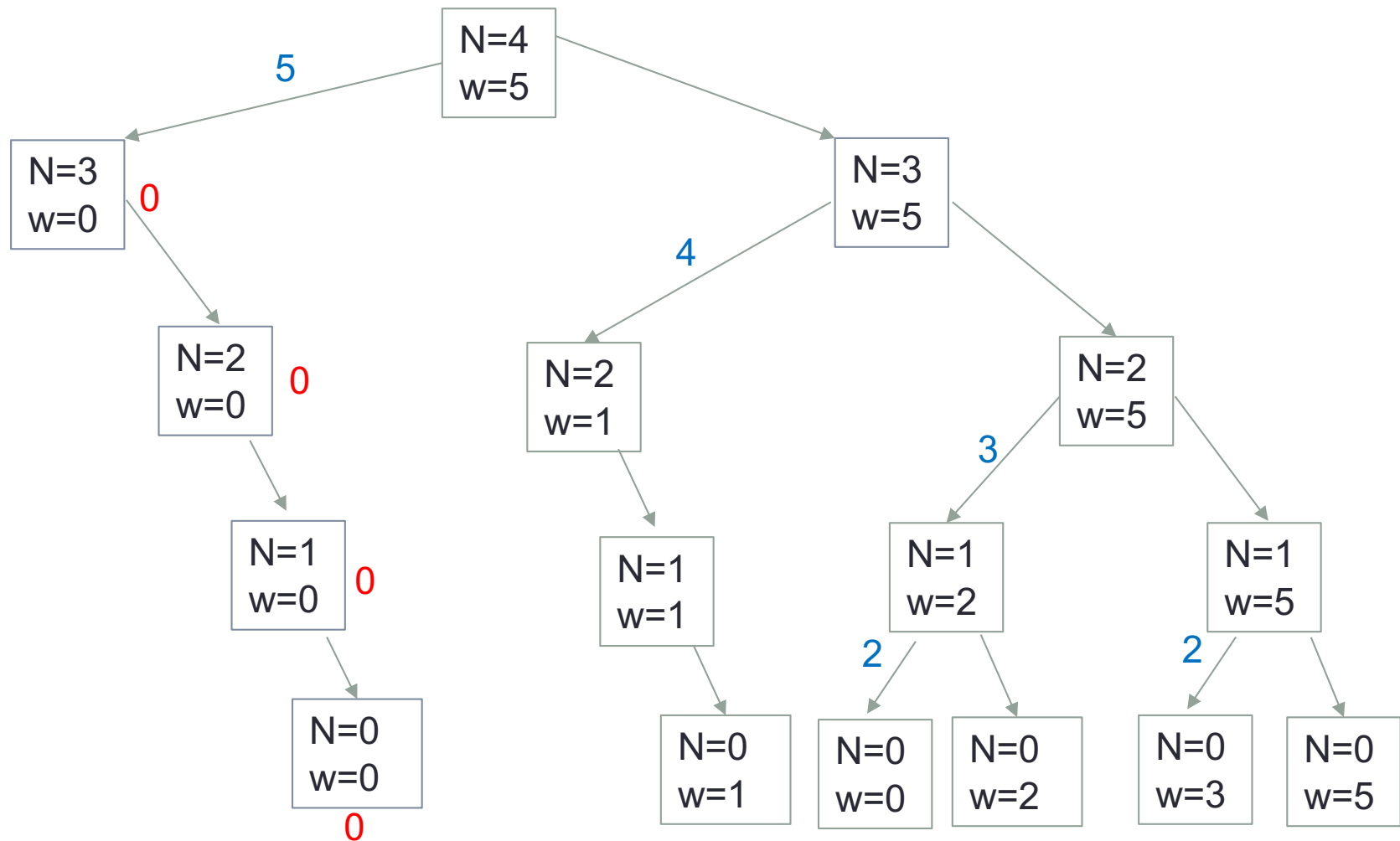
Peso

$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

$W = 2, 3, 4, 5$

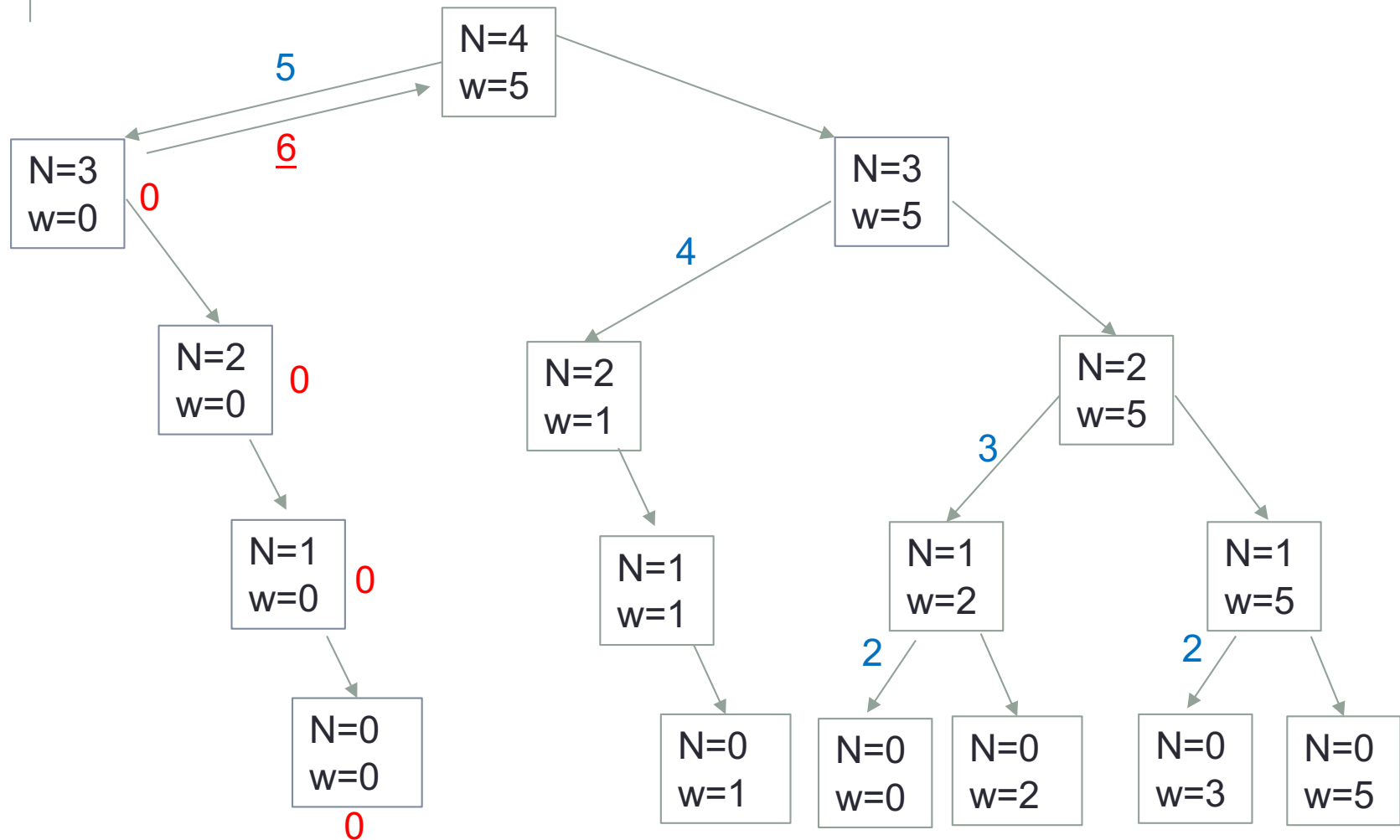
Peso

$B = 3, 4, 5, \underline{6}$

Beneficio

$N = 1, 2, 3, 4$

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

W = 2, 3, 4, 5

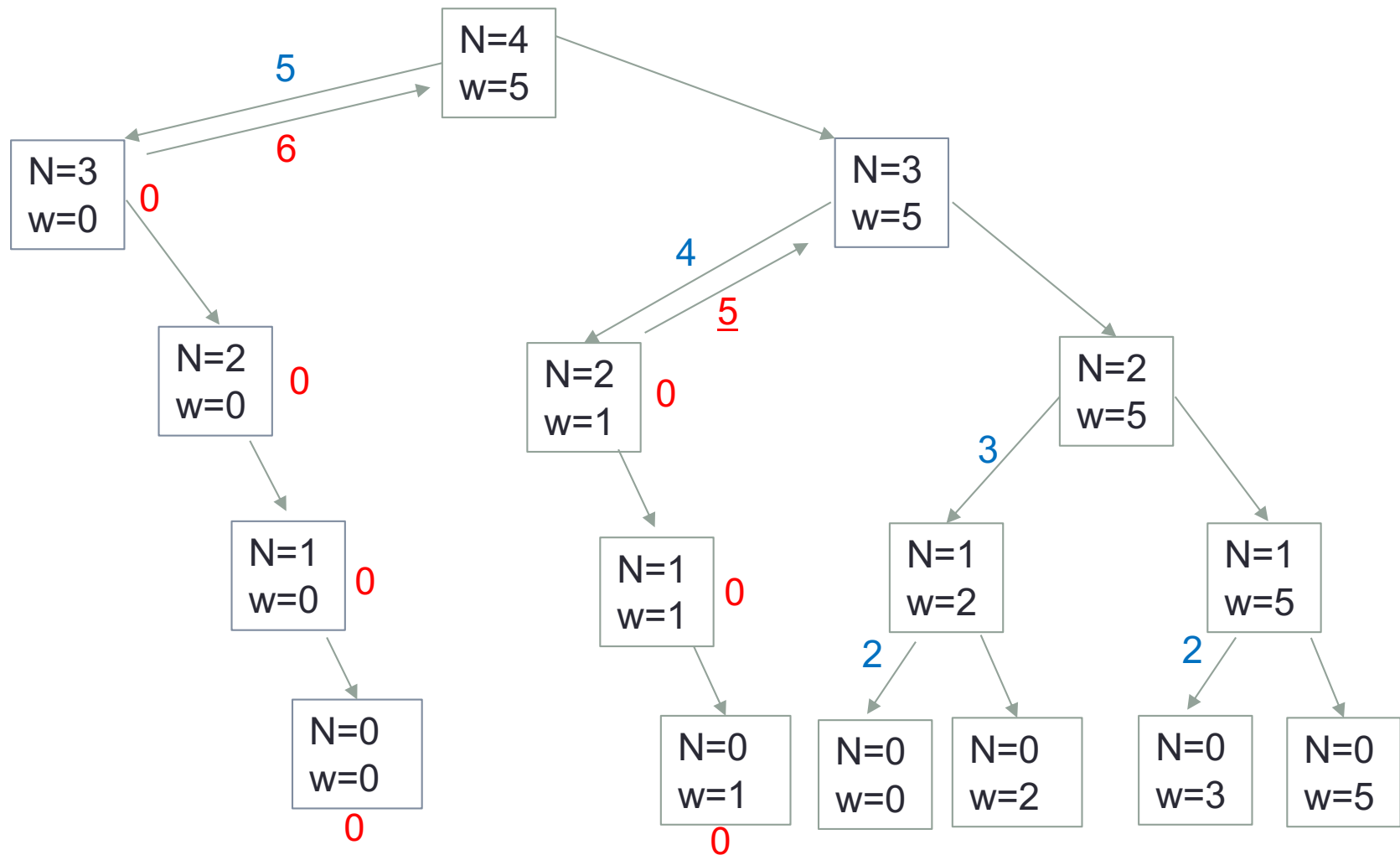
Peso

B = 3, 4, 5, 6

Beneficio

N = 1, 2, 3, 4

Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos



El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!

$W = 2, 3, 4, 5$

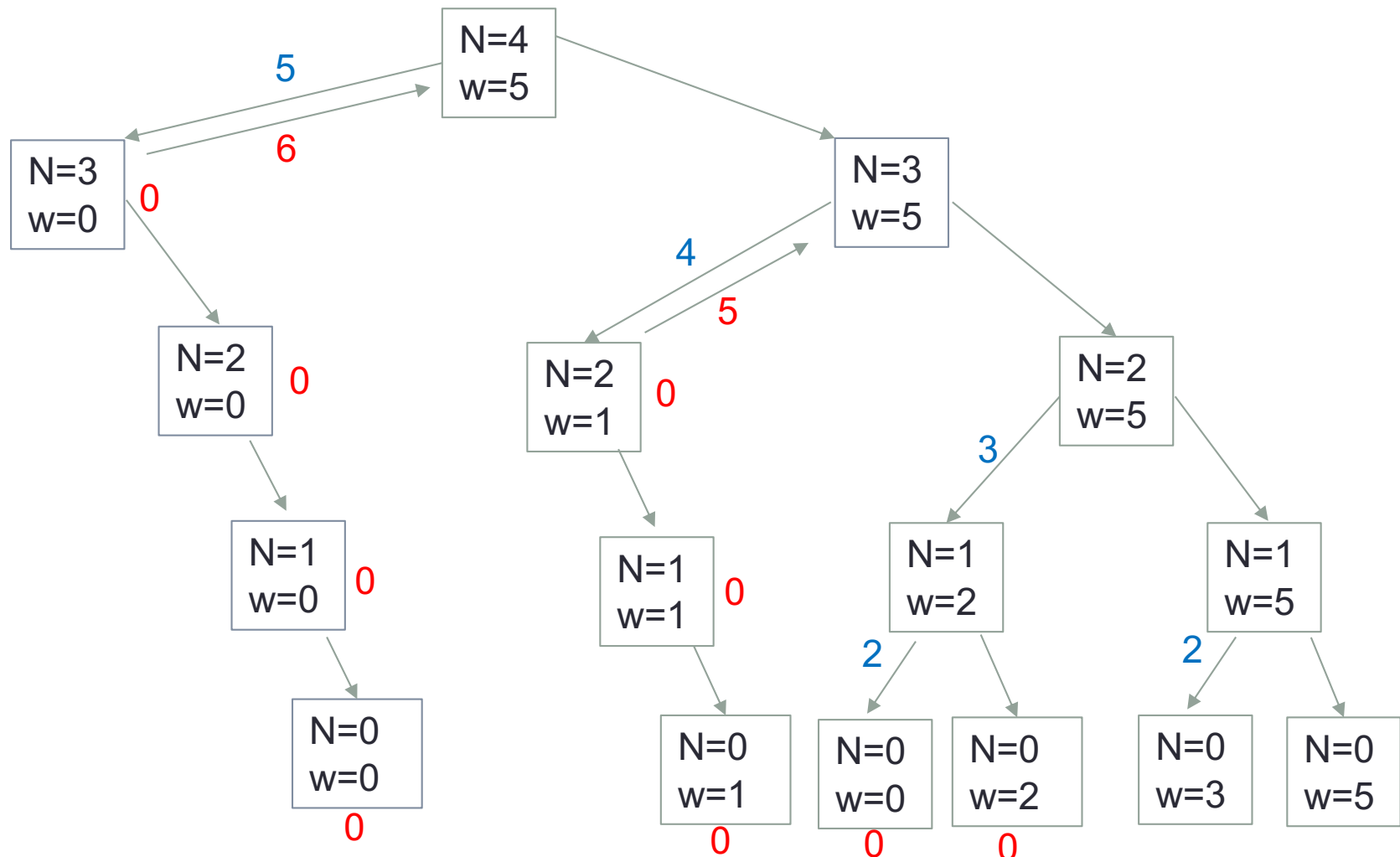
Peso

$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

$W = 2, 3, 4, 5$

Peso

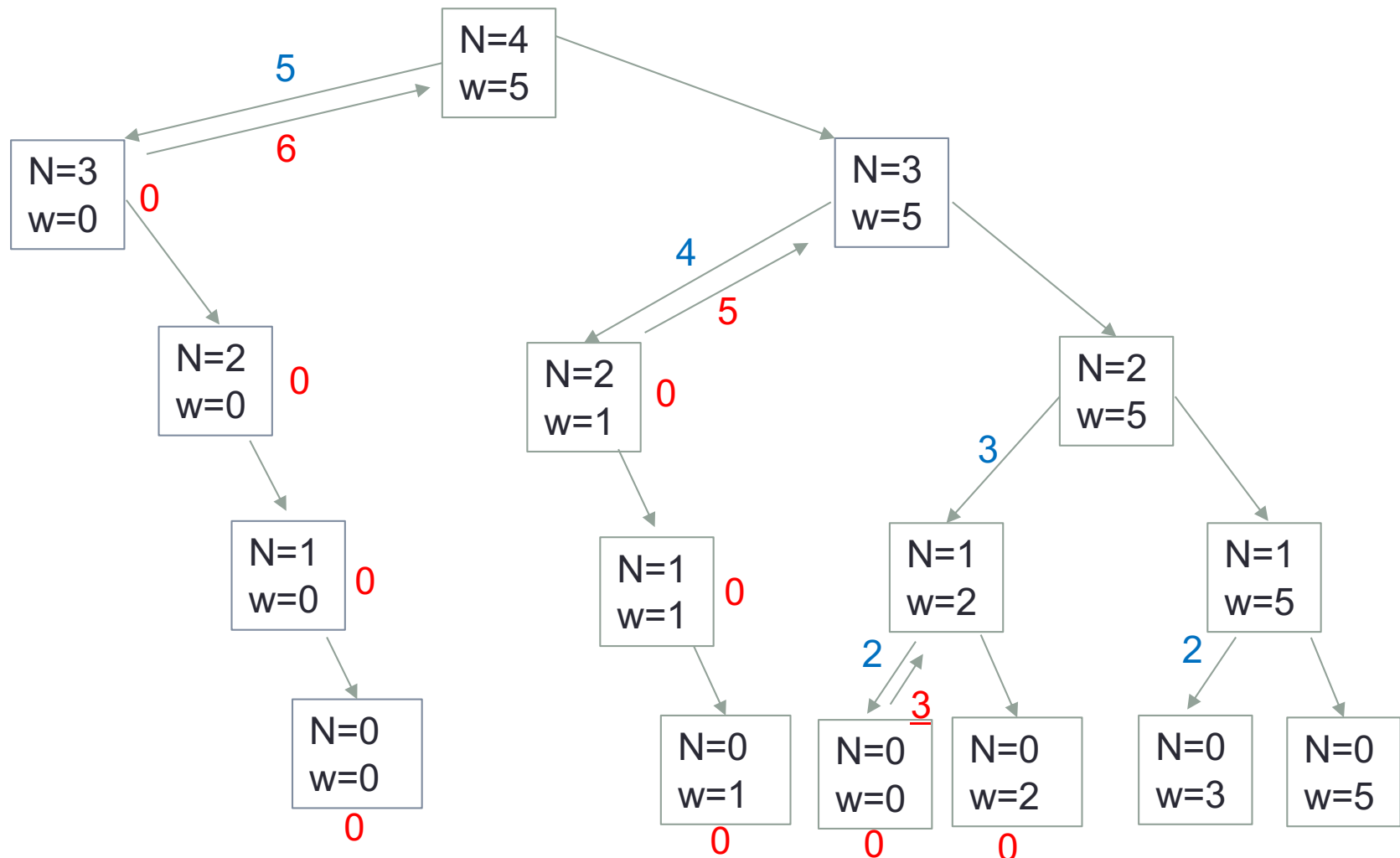
$B = \underline{3}, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

$W = 2, 3, 4, 5$

Peso

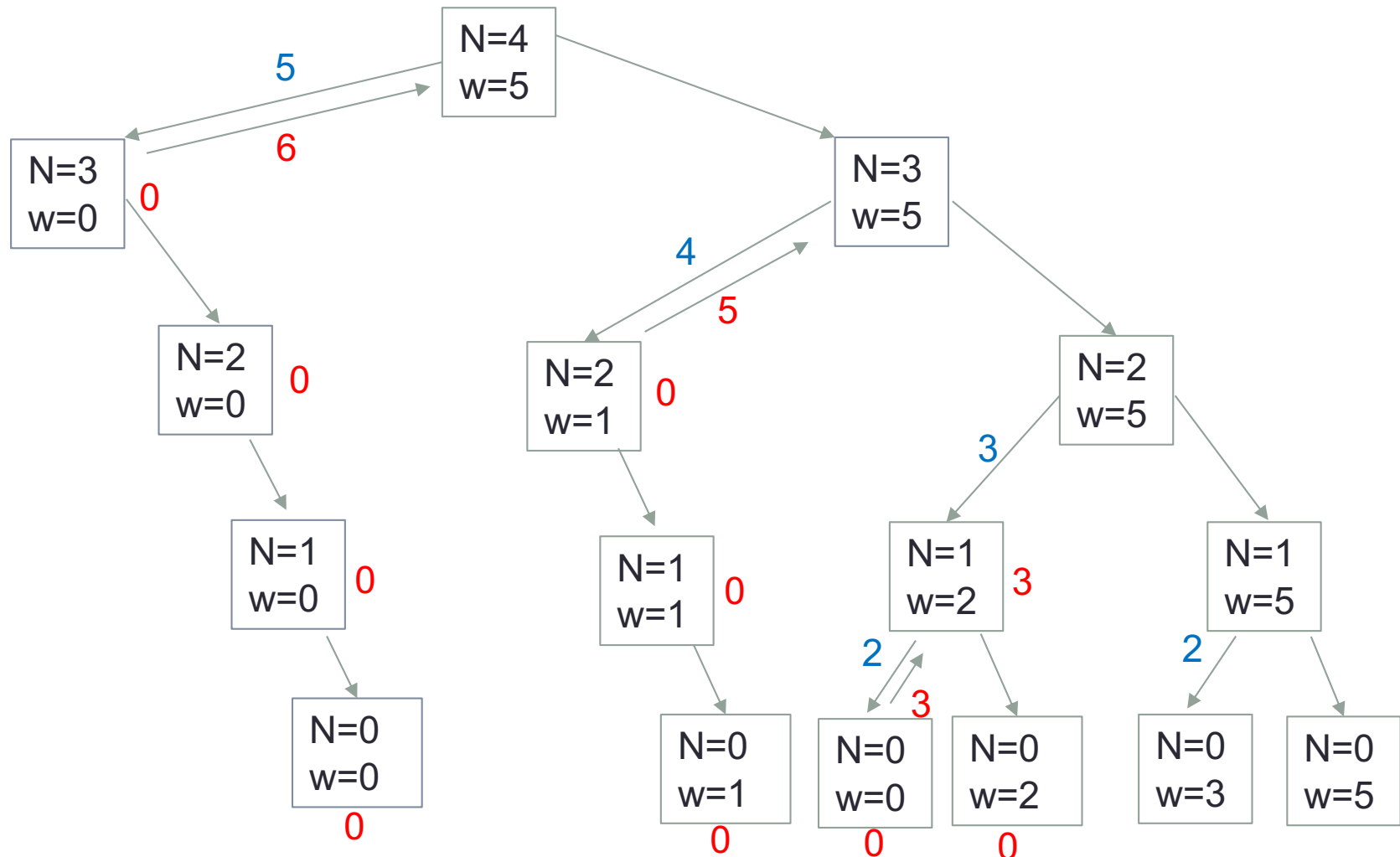
$B = \underline{3}, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*





$W = 2, 3, 4, 5$

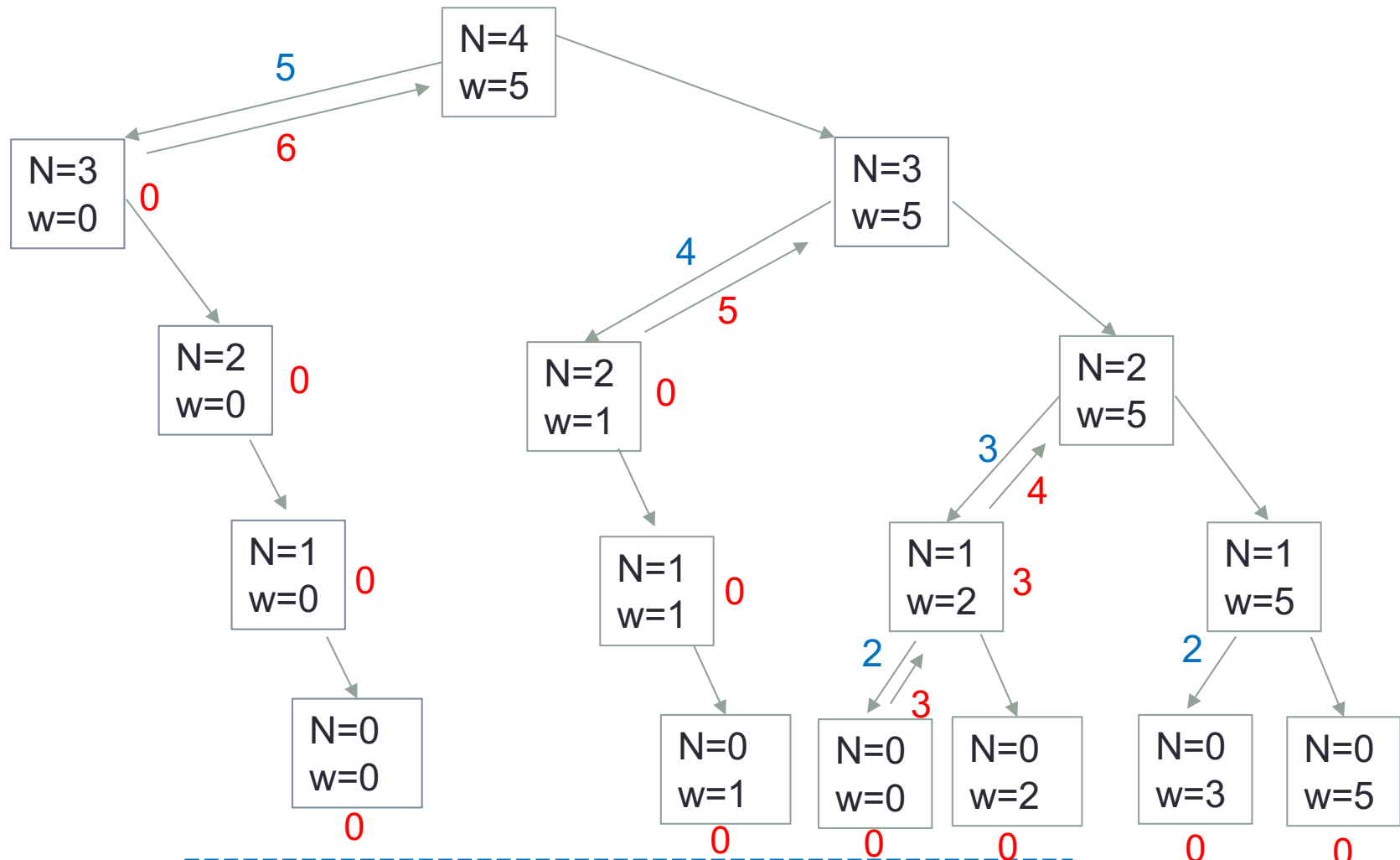
Peso

$B = 3, 4, 5, 6$

Beneficio

$N = 1, 2, 3, 4$

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

W = 2, 3, 4, 5

Peso

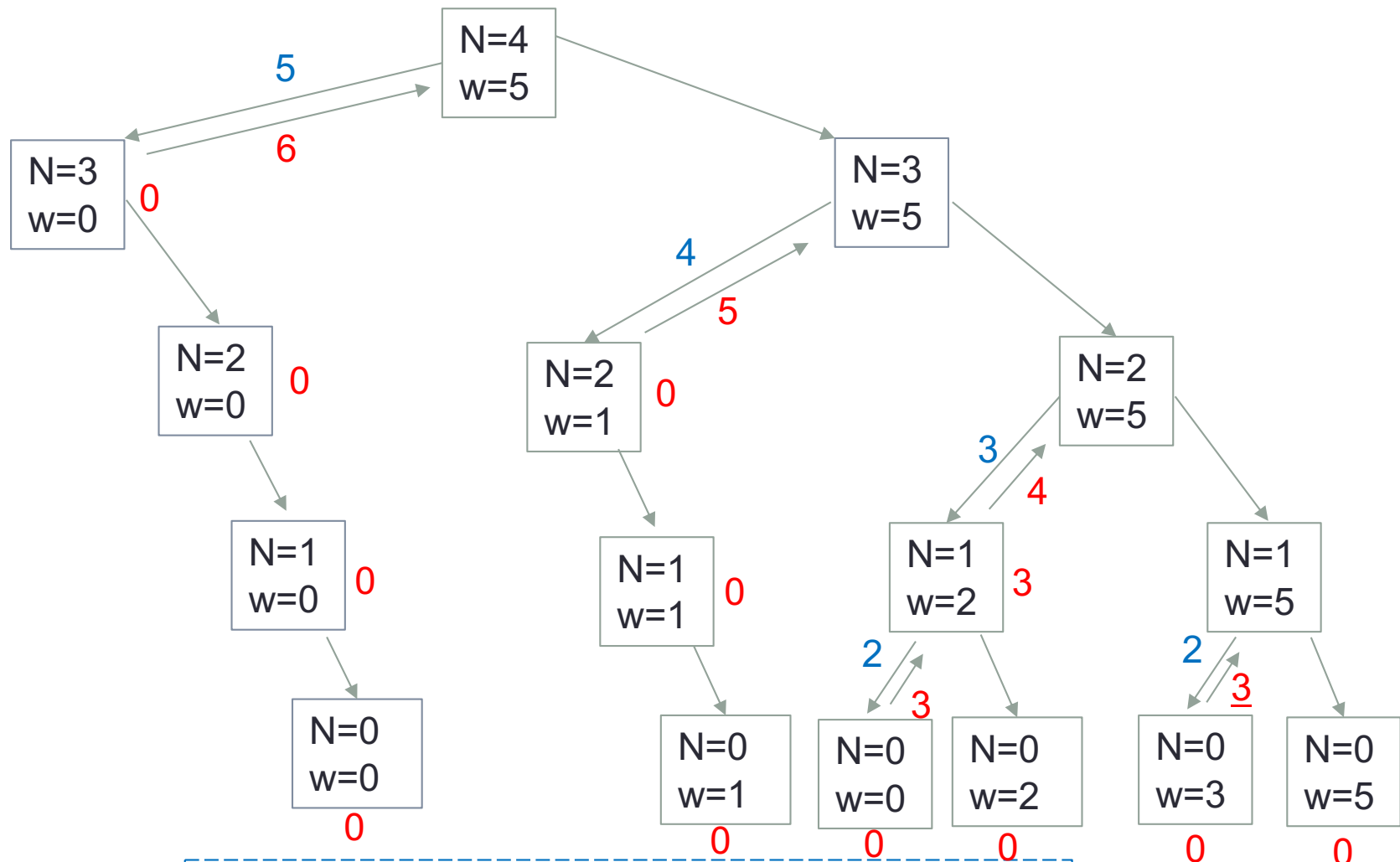
B = 3, 4, 5, 6

Beneficio

N = 1, 2, 3, 4



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

W = 2, 3, 4, 5

Peso

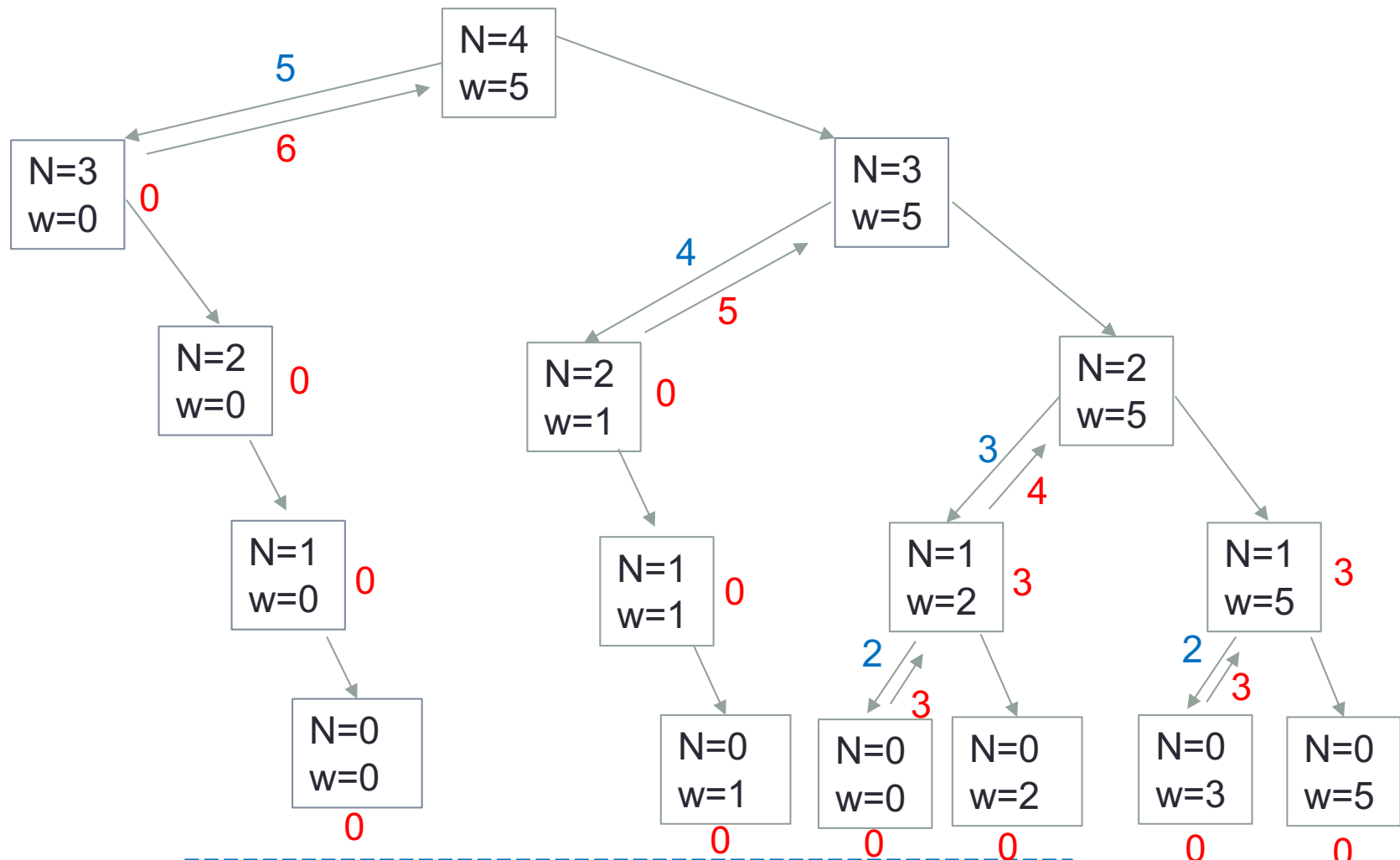
B = 3, 4, 5, 6

Beneficio

N = 1, 2, 3, 4



*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*



*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

W = 2, 3, 4, 5

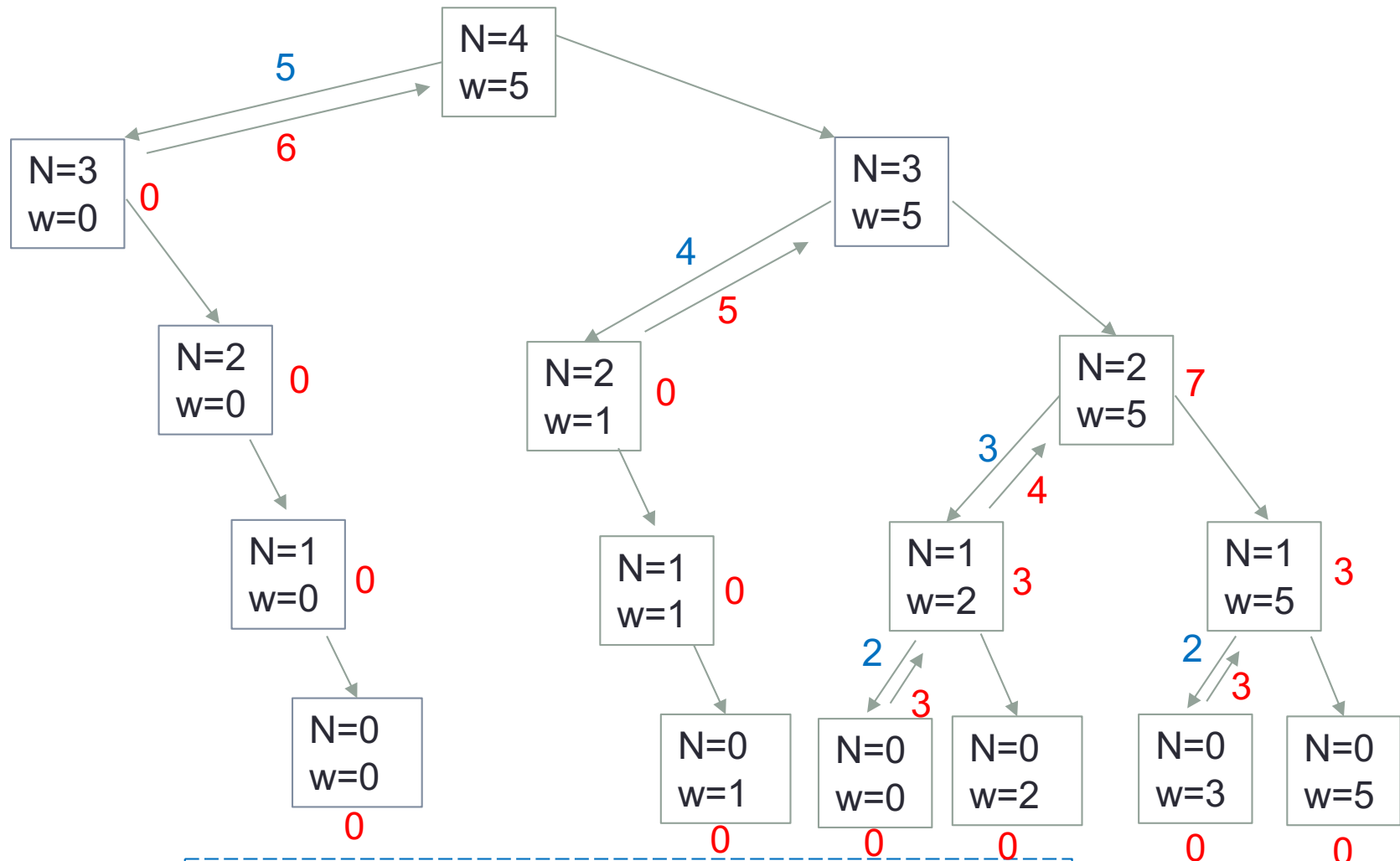
Peso

B = 3, 4, 5, 6

Beneficio

N = 1, 2, 3, 4

*Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos*

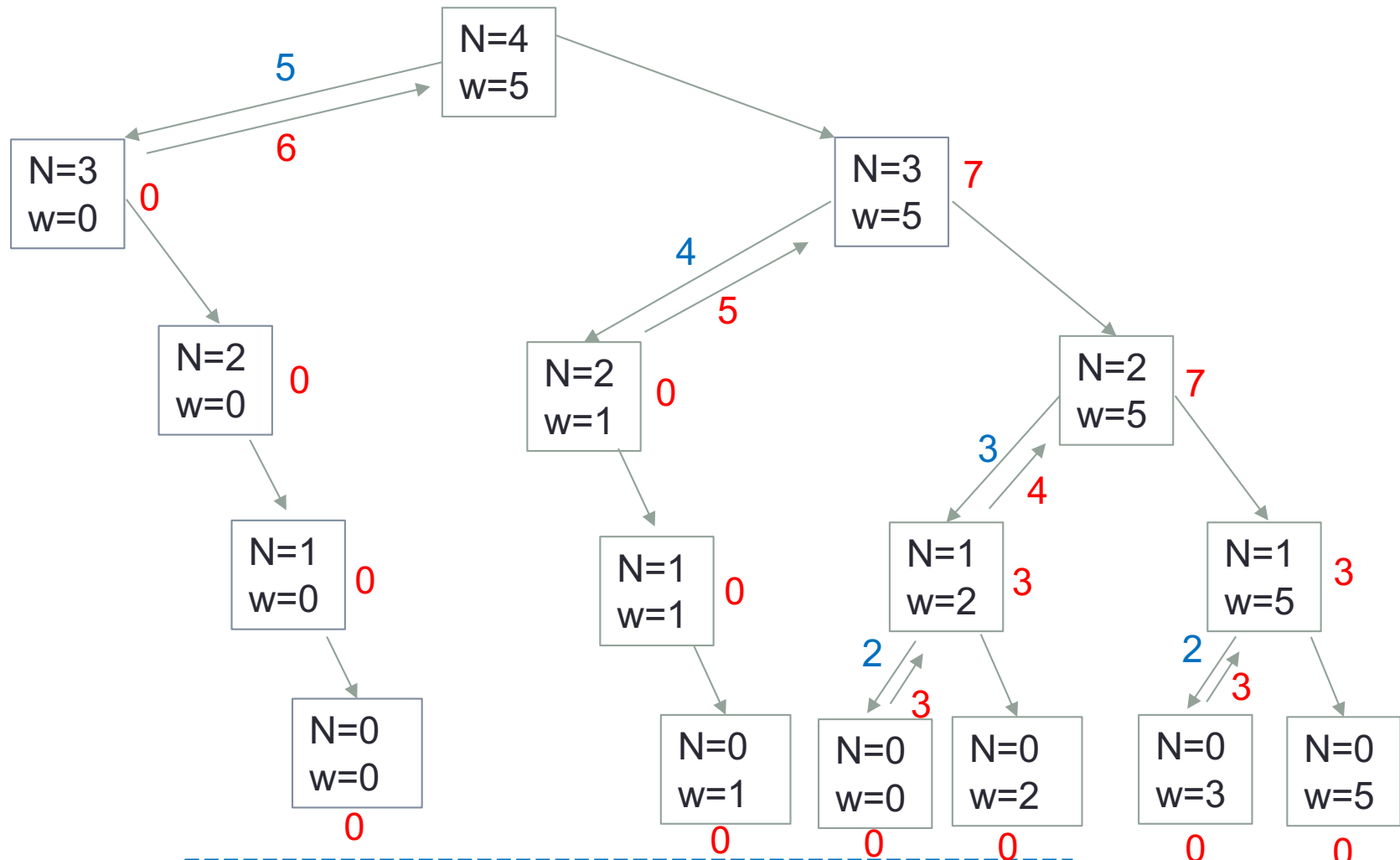


*El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!*

$W = 2, 3, 4, 5$   
 $B = 3, 4, 5, 6$   
 $N = 1, 2, 3, 4$

Peso  
 Beneficio

Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos

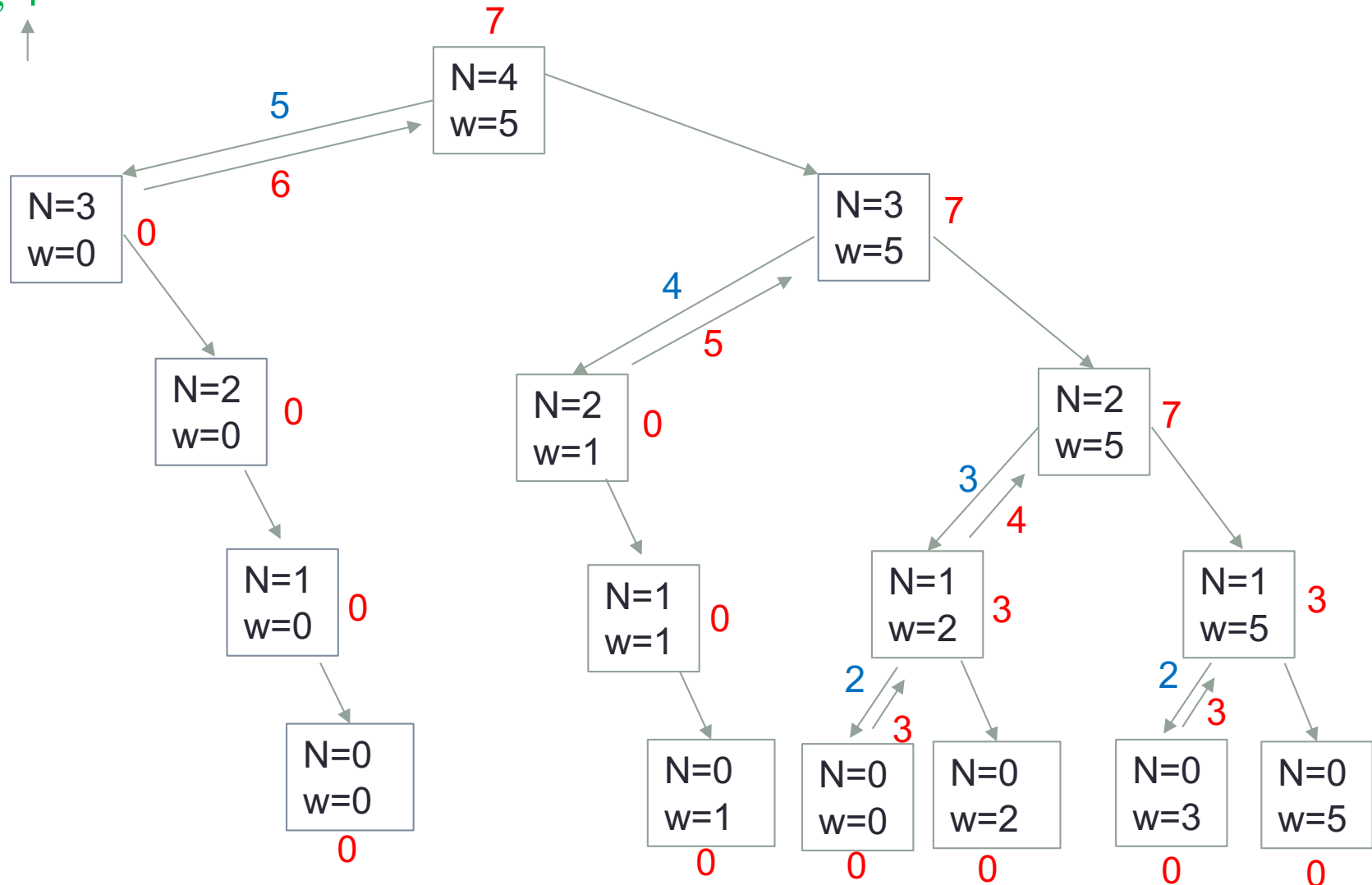


El **beneficio** de cada elección lo calculamos al volver de las llamadas recursivas!

$W = 2, 3, 4, 5$   
 $B = 3, 4, 5, 6$   
 $N = 1, 2, 3, 4$

Peso  
 Beneficio

Debemos calcular la mejor combinación que tenga un peso máximo de 5 kilos



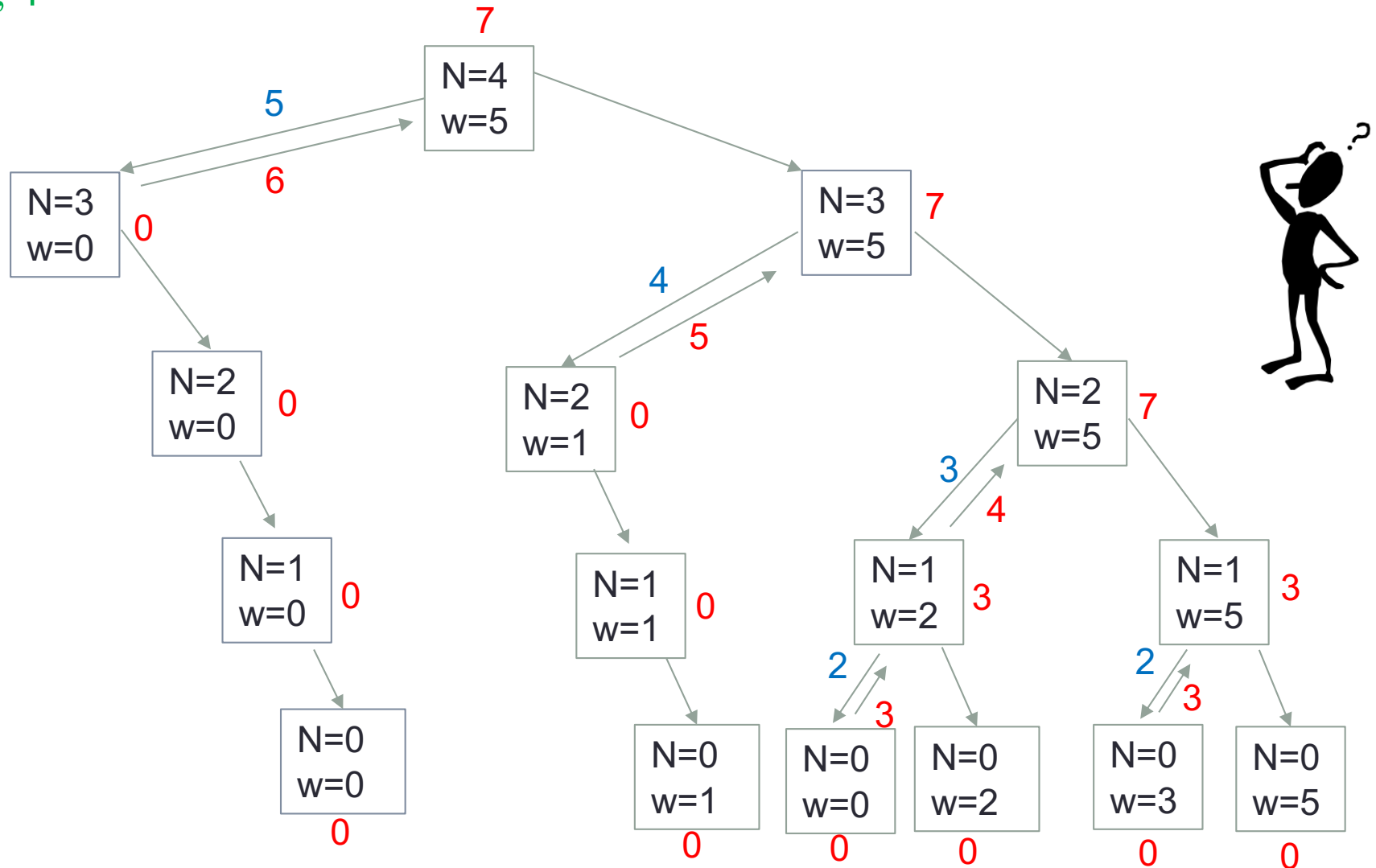
Árbol de todas las posibles llamadas recursivas con su **beneficio**

$W = 2, 3, 4, 5$

$B = 3, 4, 5, 6$

$N = 1, 2, 3, 4$

¿Cual es la ecuación de recurrencia ?

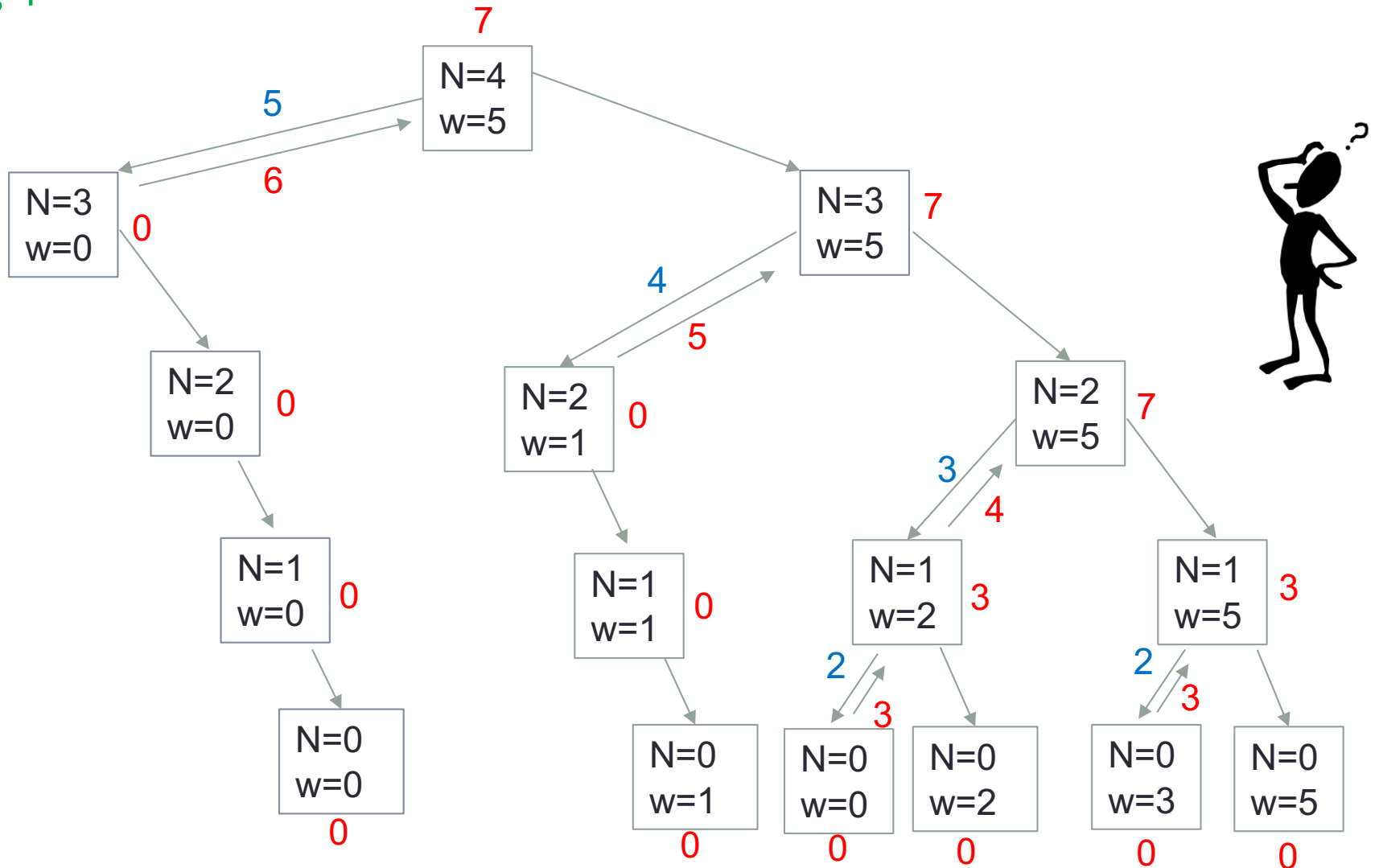


Árbol de todas las posibles llamadas recursivas con su **beneficio**



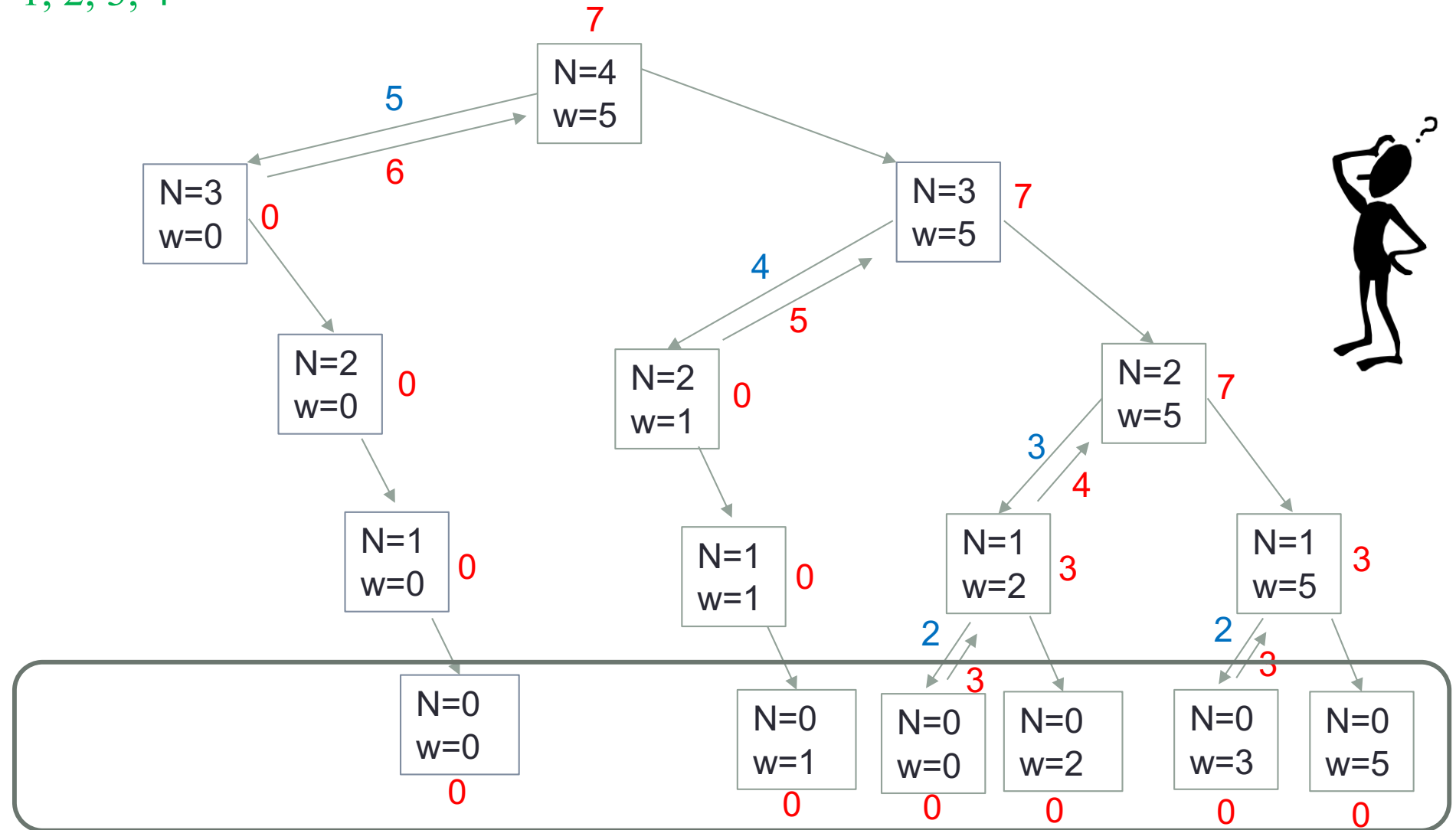
$W = 2, 3, 4, 5$   
 $B = 3, 4, 5, 6$   
 $N = 1, 2, 3, 4$

$t(n, w)$



$W = 2, 3, 4, 5$   
 $B = 3, 4, 5, 6$   
 $N = 1, 2, 3, 4$

$t(n,w)$

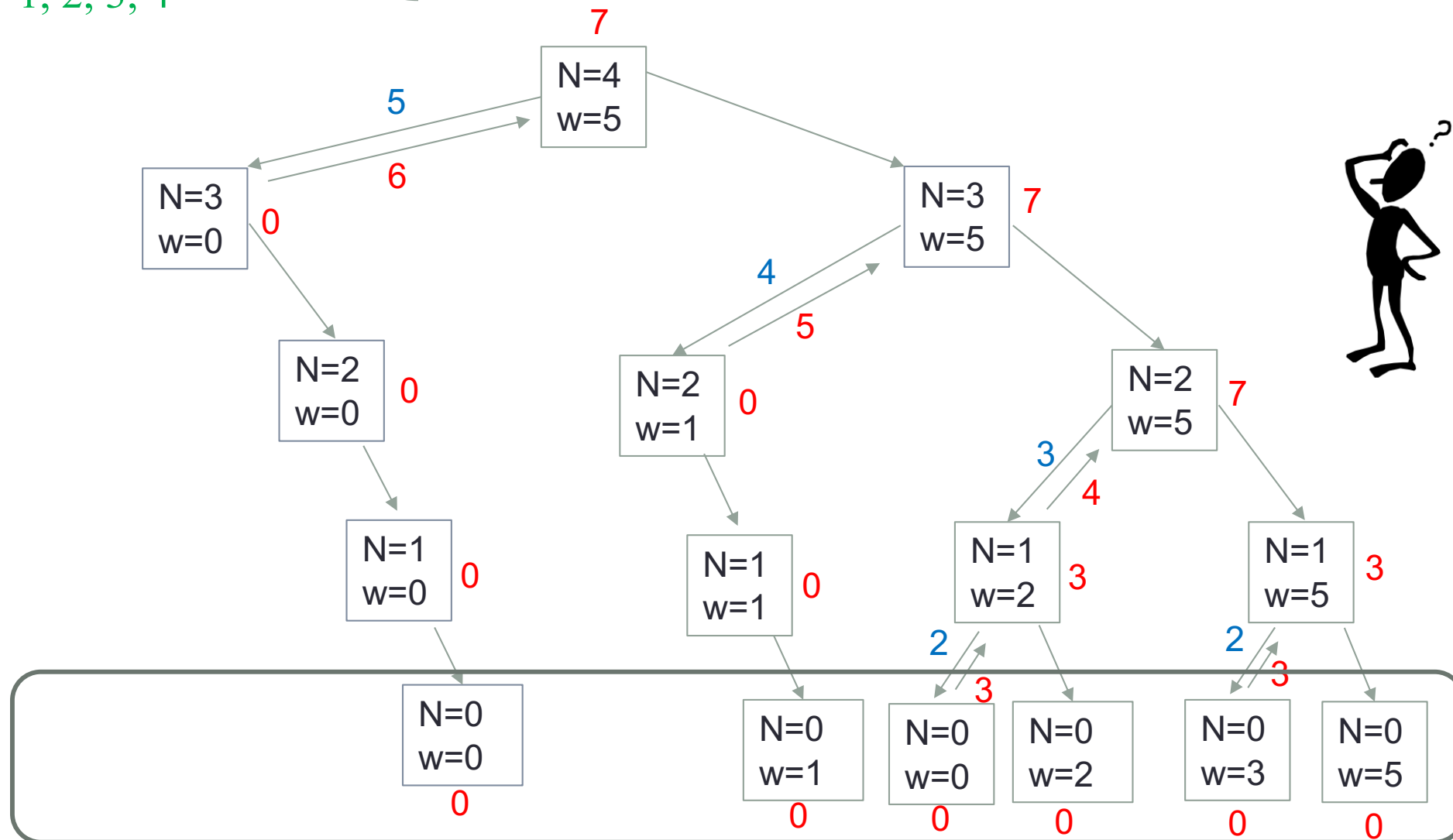


*Caso 1: No quedan items*

$W = 2, 3, 4, 5$   
 $B = 3, 4, 5, 6$   
 $N = 1, 2, 3, 4$

$t(n, w)$

$n \leq 0 \leftarrow \text{Caso Base}$



*Caso 1: No quedan items*

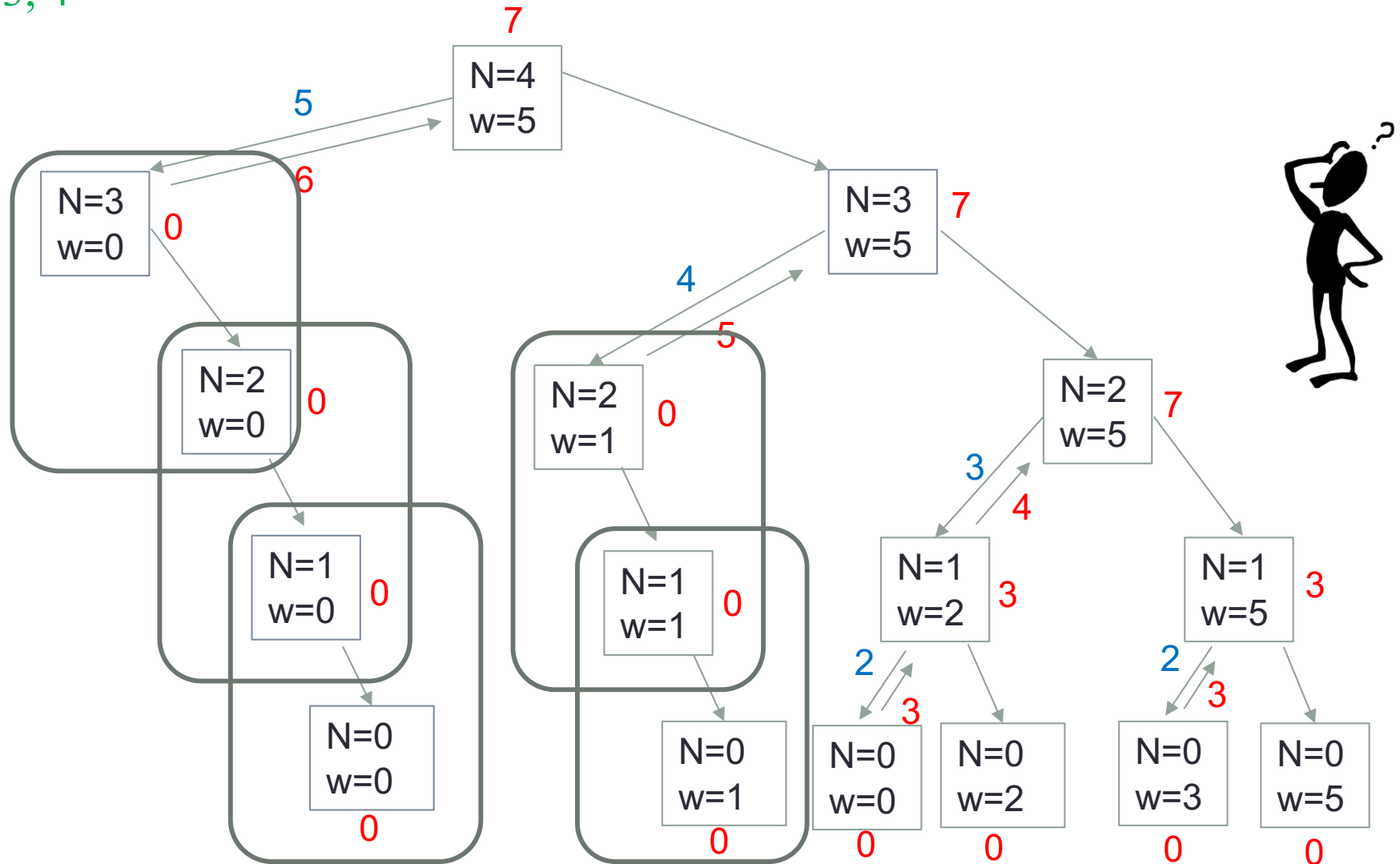
$W = 2, 3, 4, 5$

$B = 3, 4, 5, 6$

$N = 1, 2, 3, 4$

$t(n, w) \begin{cases} 0 \end{cases}$

$: n \leq 0 \leftarrow \text{Caso Base}$



*Caso 2: No puedo cogerlo*

$W = 2, 3, 4, 5$

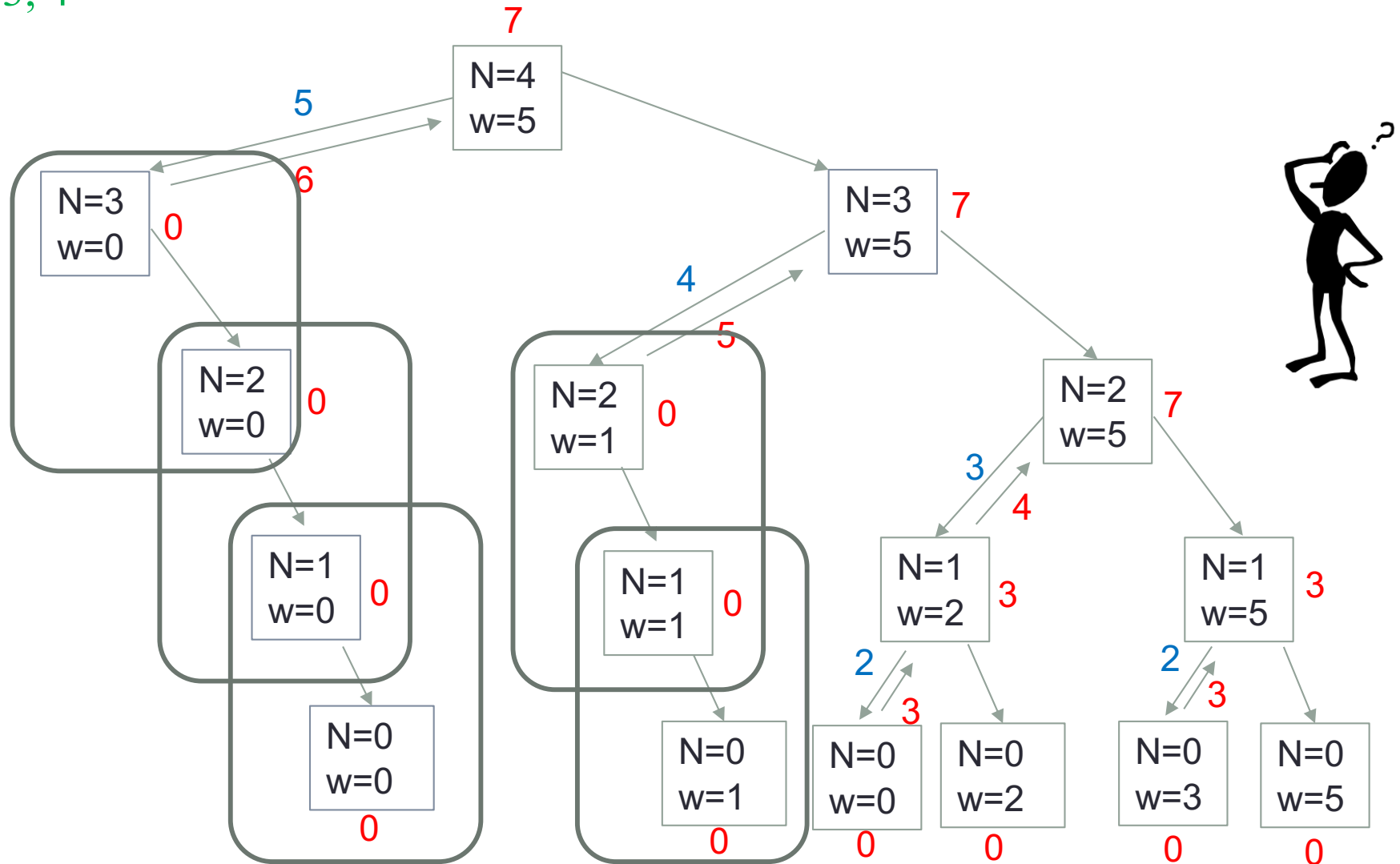
$B = 3, 4, 5, 6$

$N = 1, 2, 3, 4$

$$t(n, w) = \begin{cases} t(n-1, w) & : W_n > w \\ 0 & : n \leq 0 \end{cases}$$

$: W_n > w$

$: n \leq 0 \quad \leftarrow \text{Caso Base}$



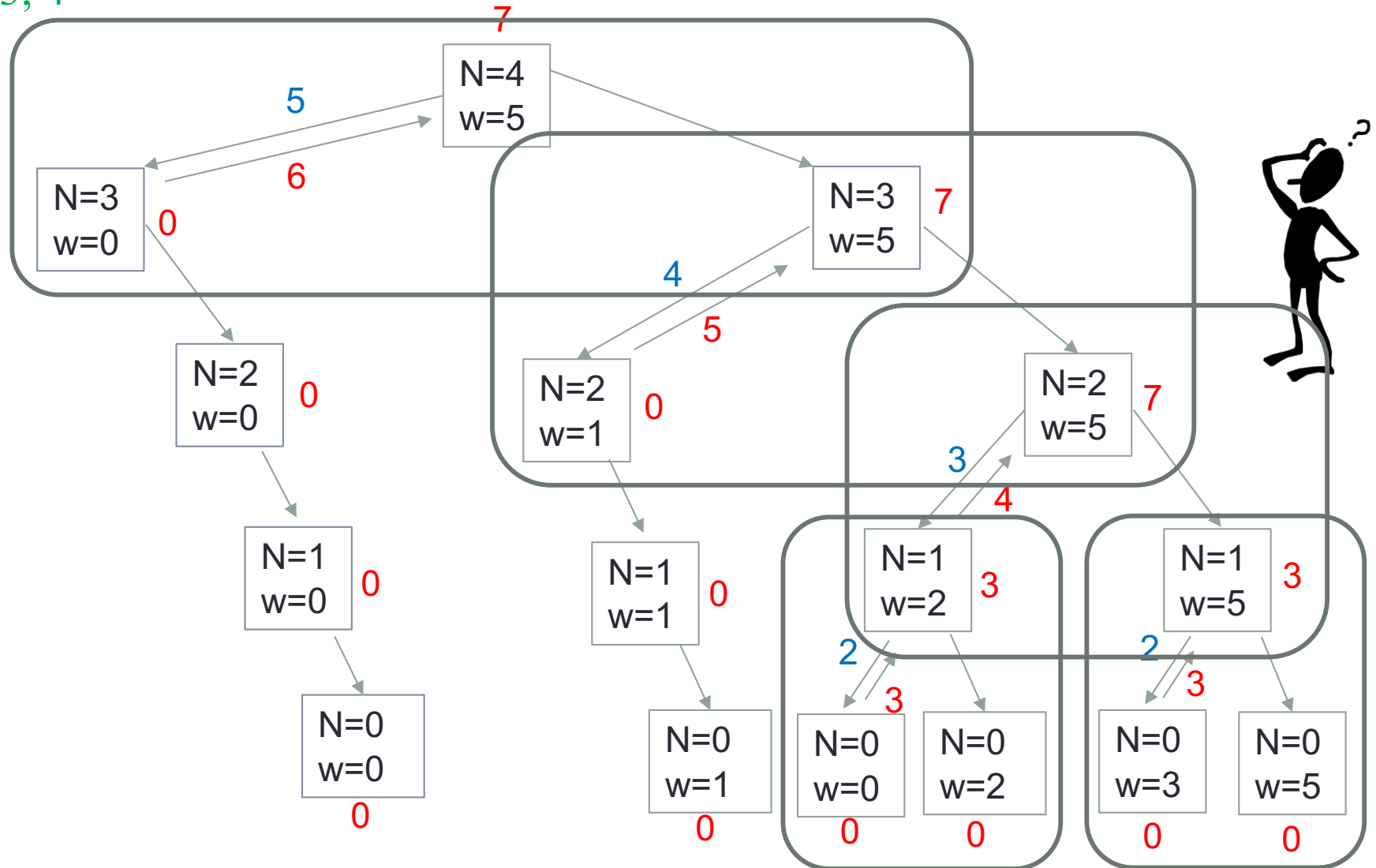
*Caso 2: No puedo cogerlo*

$W = 2, 3, 4, 5$   
 $B = 3, 4, 5, 6$   
 $N = 1, 2, 3, 4$

$$t(n, w) = \begin{cases} t(n-1, w) & : W_n > w \\ \max(t(n-1, w), t(n-1, w - W_n) + B_n) & : n \leq 0 \end{cases}$$

$: W_n > w$

$: n \leq 0 \quad \leftarrow \text{Caso Base}$

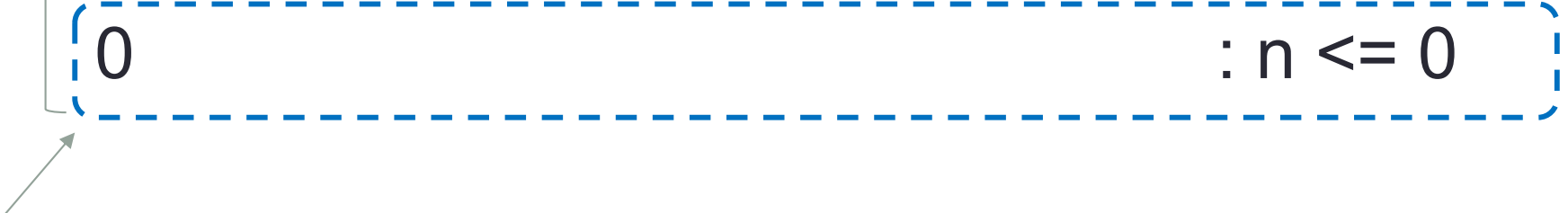


*Caso 3: Caso general*

# Lectura de la Recurrencia

$$t(n,w) \begin{cases} t(n-1, w) & : W_n > w \\ \max (t(n-1,w), t(n-1,w-W_n) + B_n) & \\ 0 & : n \leq 0 \end{cases}$$

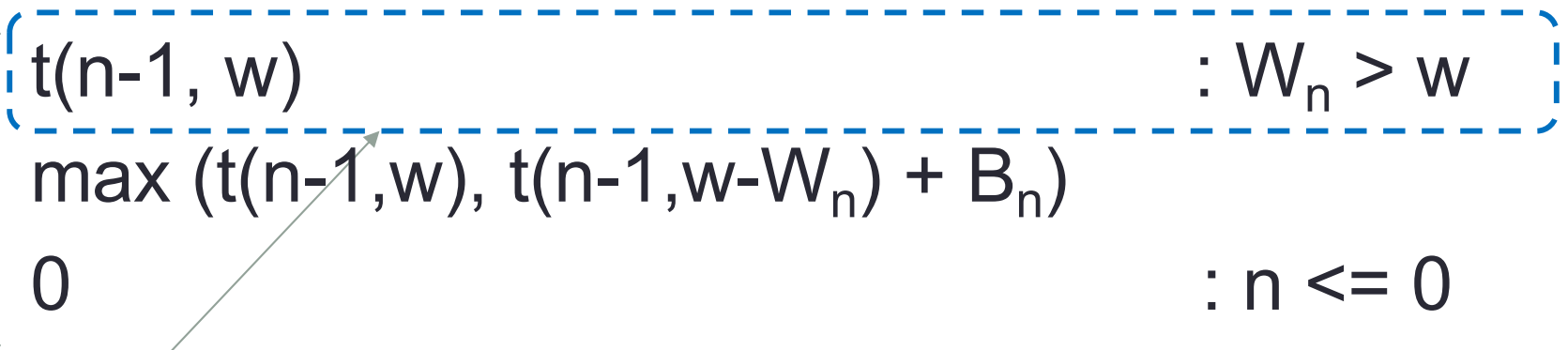
# Lectura de la Recurrencia

$$\begin{array}{lcl}
 t(n,w) \left\{ \begin{array}{l} t(n-1, w) \\ \max (t(n-1,w), t(n-1,w-W_n) + B_n) \\ 0 \end{array} \right. & \begin{array}{l} : W_n > w \\ \\ : n \leq 0 \end{array}
 \end{array}$$


El beneficio máximo que puedo obtener cuando no me queda ningún elemento es 0

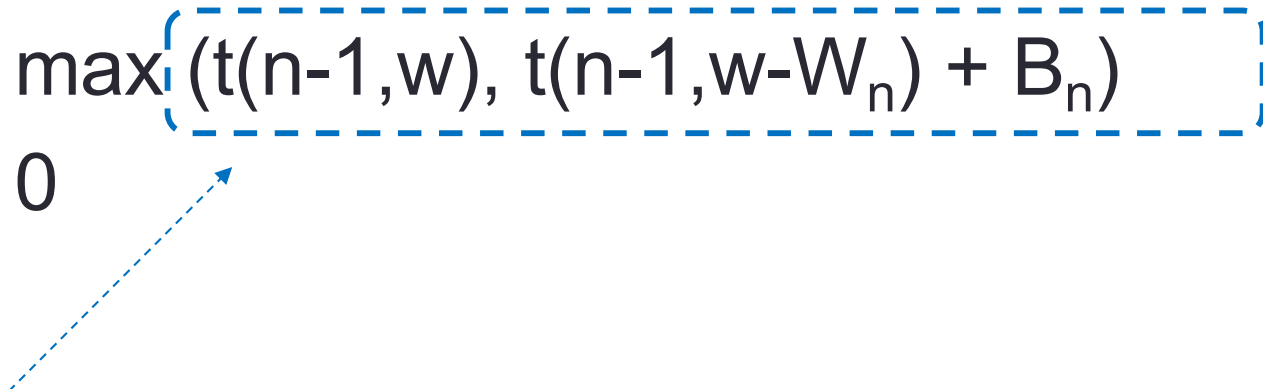


# Lectura de la Recurrencia

$$t(n, w) \begin{cases} \boxed{t(n-1, w)} & : W_n > w \\ \max(t(n-1, w), t(n-1, w - W_n) + B_n) & \\ 0 & : n \leq 0 \end{cases}$$


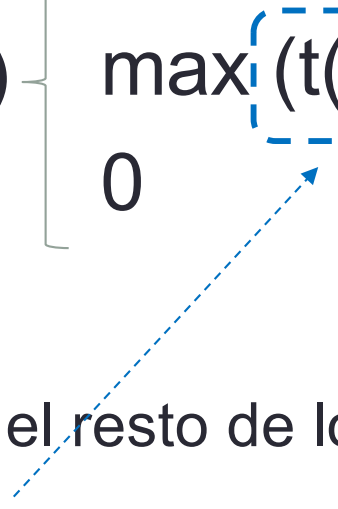
Si el peso del elemento  $n$  excede el peso que puedo añadir a la mochila, no podemos añadirlo y pasamos al siguiente

# Lectura de la Recurrencia

$$t(n, w) \begin{cases} t(n-1, w) & : W_n > w \\ \max \{ t(n-1, w), t(n-1, w - W_n) + B_n \} & : n \leq 0 \\ 0 & \end{cases}$$


En el resto de los casos, tenemos dos posibilidades:

# Lectura de la Recurrencia

$$t(n, w) \begin{cases} t(n-1, w) & : W_n > w \\ \max \{ \boxed{t(n-1, w)}, t(n-1, w - W_n) + B_n \} & \\ 0 & : n \leq 0 \end{cases}$$


En el resto de los casos, tenemos dos posibilidades:

1. No añadir el elemento  $n$ , con lo que el problema se reduce a calcular la mochila óptima sin tener en cuenta  $n$

# Lectura de la Recurrencia

$$t(n, w) \begin{cases} t(n-1, w) & : W_n > w \\ \max(t(n-1, w), t(n-1, w - W_n) + B_n) & : n \leq 0 \\ 0 \end{cases}$$

En el resto de los casos, tenemos dos posibilidades:

1. No añadir el elemento  $n$ , con lo que el problema se reduce a calcular la mochila óptima sin tener en cuenta  $n$
2. Añadir el elemento  $n$ , con lo que ahora tenemos que calcular la mochila óptima restando su peso ( $w - W_n$ ) pero teniendo en cuenta su beneficio ( $B_n$ )

$W = 2, 3, 4, 5$

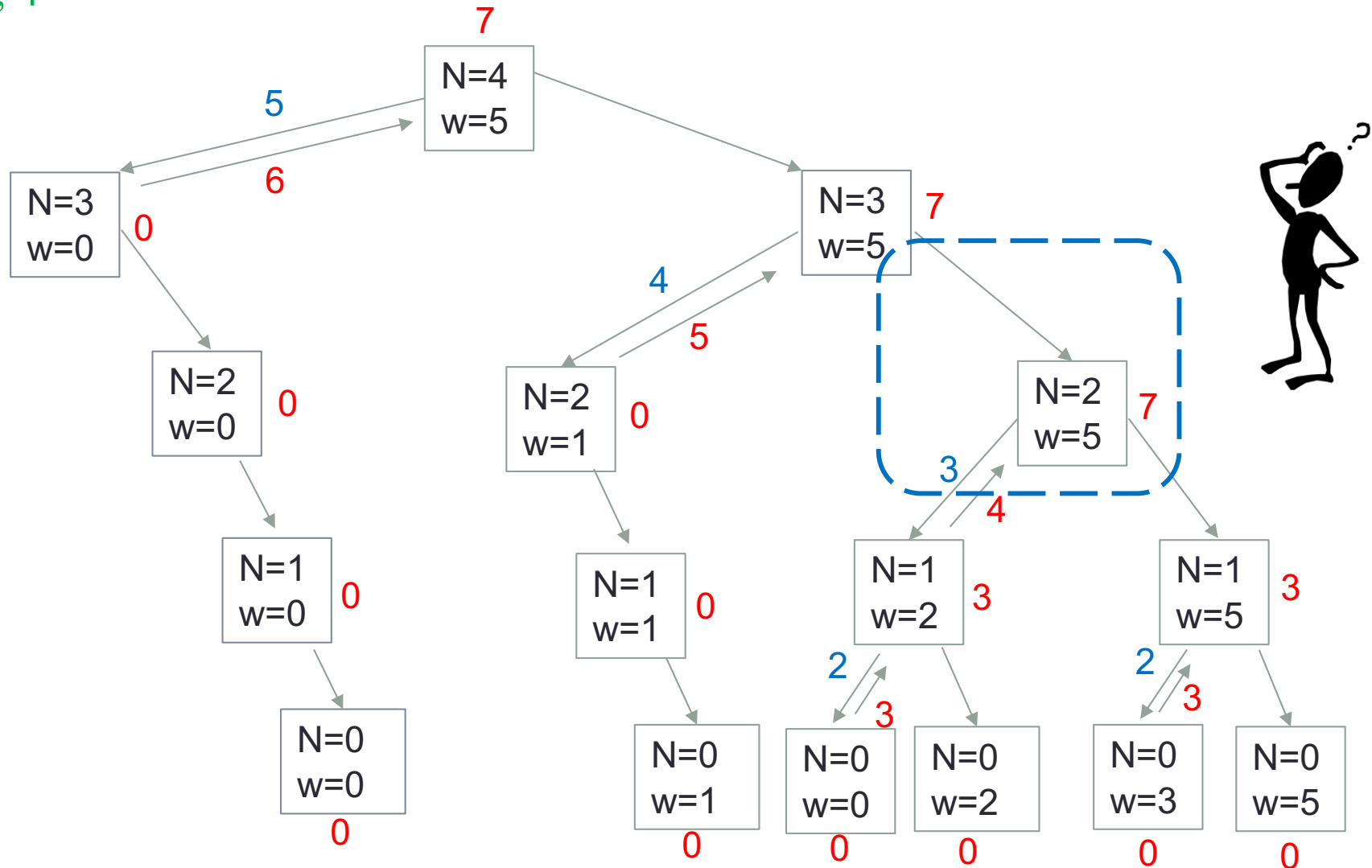
$B = 3, 4, 5, 6$

$N = 1, 2, 3, 4$

$$t(n, w) = \begin{cases} t(n-1, w) & : W_n > w \\ \max(t(n-1, w), t(n-1, w - W_n) + B_n) & : n \leq 0 \end{cases}$$

$: W_n > w$

$: n \leq 0 \leftarrow \text{Caso Base}$



*Caso 3: Caso general*

$W = 2, 3, 4, 5$

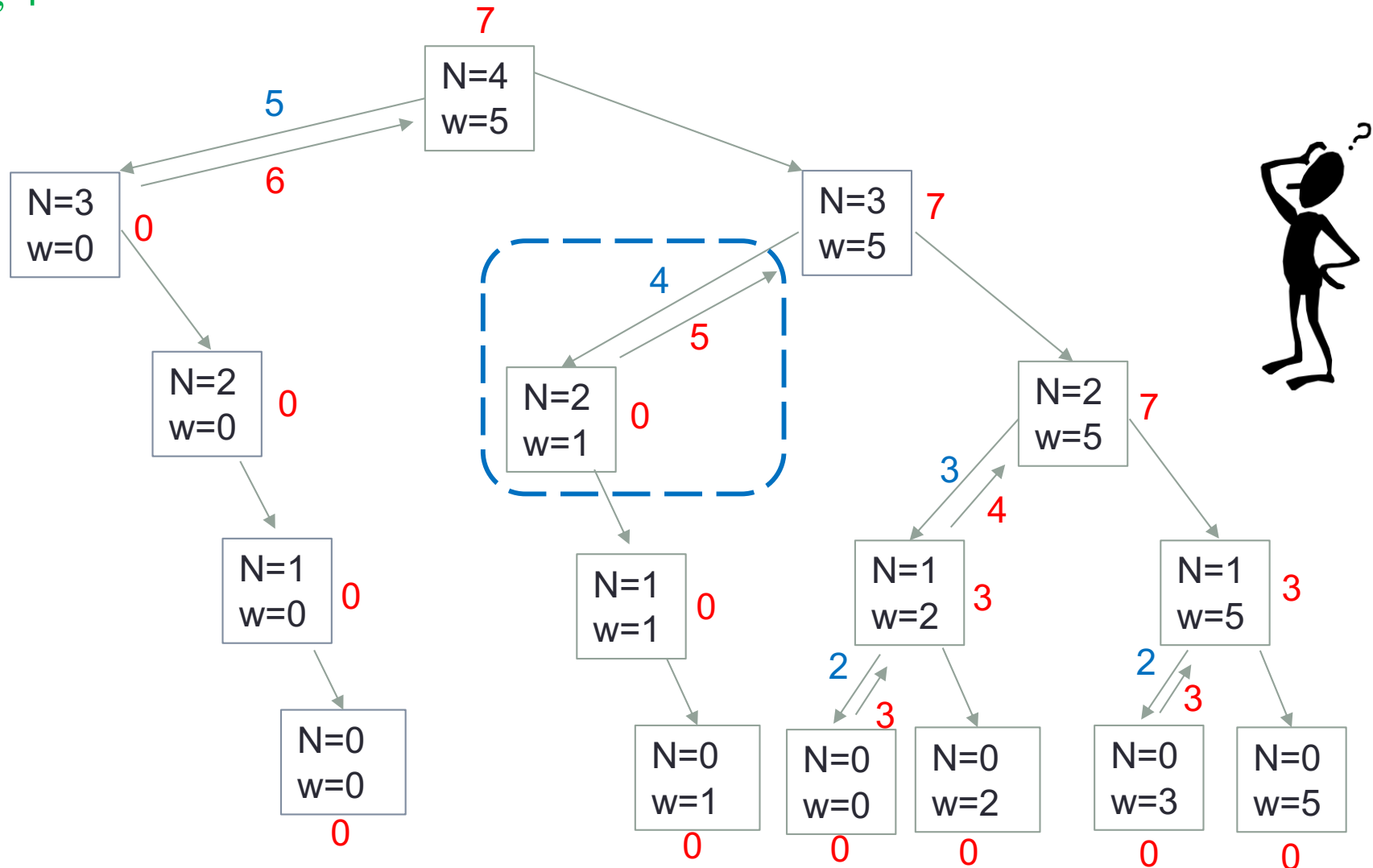
$B = 3, 4, 5, 6$

$N = 1, 2, 3, 4$

$$t(n, w) = \begin{cases} t(n-1, w) & : W_n > w \\ \max(t(n-1, w), \boxed{t(n-1, w - W_n) + B_n}) & : n \leq 0 \end{cases}$$

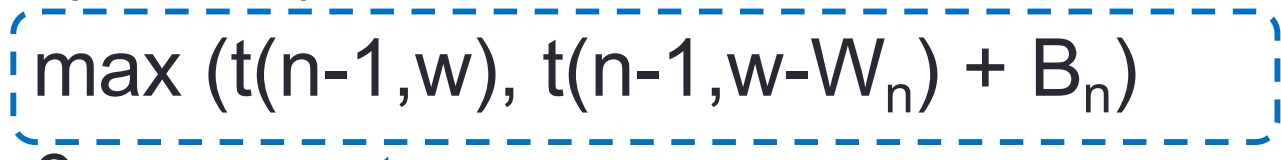
$: W_n > w$

$: n \leq 0 \leftarrow \text{Caso Base}$



*Caso 3: Caso general*

# Lectura de la Recurrencia

$$t(n, w) \begin{cases} t(n-1, w) & : W_n > w \\ \max(t(n-1, w), t(n-1, w - W_n) + B_n) & : n \leq 0 \\ 0 & \end{cases}$$


En el resto de los casos, tenemos dos posibilidades:

1. No añadir el elemento  $n$ , con lo que el problema se reduce a calcular la mochila óptima sin tener en cuenta  $n$
2. Añadir  $n$ , con lo que ahora tenemos que calcular la mochila óptima restando su peso ( $w - W_n$ ) pero teniendo en cuenta su beneficio ( $B_n$ )

*Y nos quedamos con la opción que maximice nuestro beneficio!*