

A
5
2
7
1
6
3
9
4
8

B

pb  
pb

A
7
1
6
3
9
4
8

B
2
5

rb  
pb

A
1
6
3
9
4
8

B
7
5
2

pb

A
6
3
9
4
8

B
1
7
5
2

rb  
rb  
pb

A
3
9
4
8

B
6
5
2
1
7

rrr  
pb

A
3
9
4

B
8
7
6
5
2
1

CUANDO QUEDAN 3 NÚMEROS,  
LLAMAMOS LA FUNCIÓN DE  
ORDENAR 3 NÚMEROS

ra  
sa

A
3
4
9

B
8
7
6
5
2
1

# Calcular coste:

Si el número a mover es menor que el MIN o mayor que el MAX del stack\_b, debemos colocar el MAX en la parte superior.

En este caso queremos mover el 7, por lo que deberemos tener el 5 (MAX) en la parte superior del stack\_b.

A	B
7	2
1	5
6	
3	
9	
4	
8	

CALCULAMOS TODAS LAS POSIBILIDADES:

Mover el 7 arriba con ra -> ra = 0		Mover el 5 arriba con rb -> rb = 1
Nos quedamos con el que tenga la mayor cantidad de movimientos (1 + 1pb)		
Si ra >= rb -> coste = ra (+1pb)		Si rb >= ra -> coste = rb (+1pb)
Mover el 7 arriba con rra -> rra = 0		Mover el 5 arriba con rrb -> rrb = 1
Nos quedamos con el que tenga la mayor cantidad de movimientos (1 + 1pb)		
Si rra >= rrb -> coste = rra (+1pb)		Si rrb >= rra -> coste = rrb (+1pb)
Mover el 7 arriba con ra -> ra = 0		Mover el 5 arriba con rrb -> rrb = 1
Sumamos ra + rrb (1 + 1pb)		
Mover el 7 arriba con rra -> rra = 0		Mover el 5 arriba con rb -> rb = 1
Sumamos rra + rb (1 + 1pb)		

# Calcular coste:

Si el número a mover NO es menor que el MIN ni mayor que el MAX del stack\_b, debemos colocar el número directamente menor al número que queremos mover en la parte superior.

En este caso, queremos mover el 3, por lo que deberemos tener el 2 en la parte superior del stack\_b.

A	B
3	6
9	5
4	2
8	1
	7

CALCULAMOS NÚMERO DEL STACK B QUE DEBEMOS PONER ARRIBA:

Cogemos el valor de order del número 3. 3->order

Generamos  $i = 1$ ;

1. Recorremos el stack\_b hasta encontrar un número cuyo order sea el 3->order - i
  2. Si no se encuentra incrementamos  $i. i++$ ;
- Repetimos los puntos 1 y 2 hasta localizar el nodo.

A	B
3	6
9	5
4	2
8	1
	7

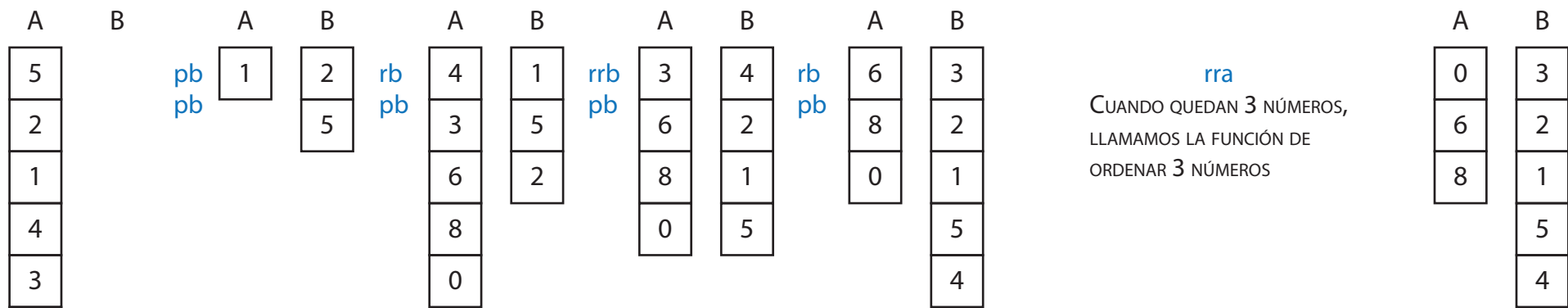
CALCULAMOS TODAS LAS POSIBILIDADES:

Mover el 3 arriba con $ra \rightarrow ra = 0$		Mover el 2 arriba con $rb \rightarrow rb = 2$
Nos quedamos con el que tenga la mayor cantidad de movimientos (2 + 1pb)		
Si $ra \geq rb \rightarrow \text{coste} = ra (+1pb)$		Si $rb \geq ra \rightarrow \text{coste} = rb (+1pb)$

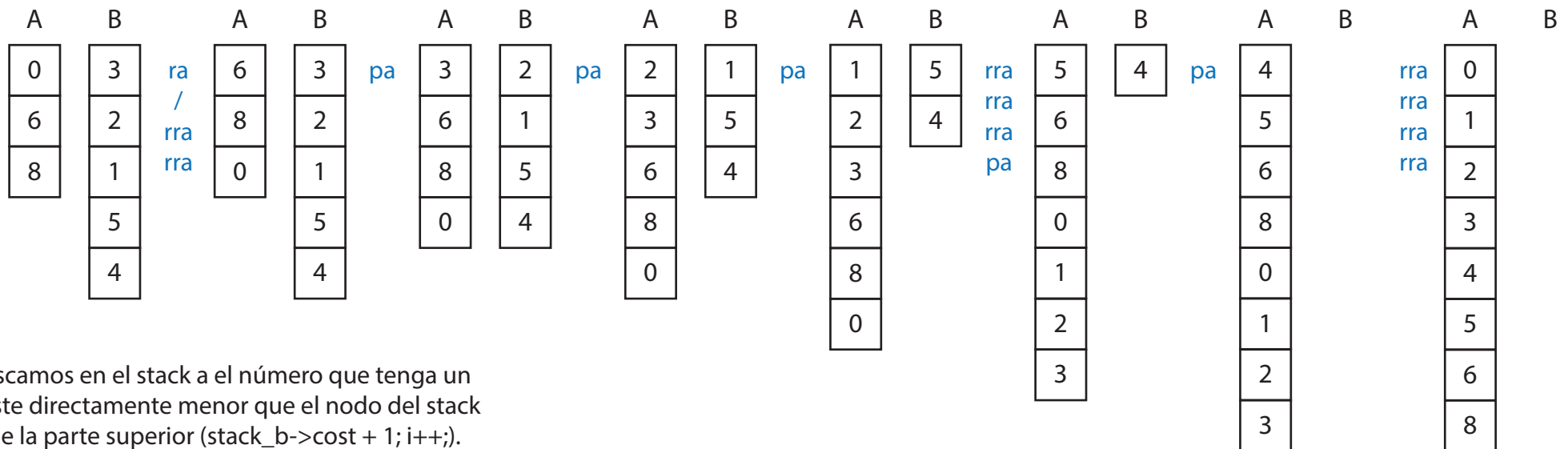
Mover el 3 arriba con $rra \rightarrow rra = 0$		Mover el 2 arriba con $rrb \rightarrow rrb = 3$
Nos quedamos con el que tenga la mayor cantidad de movimientos (3 + 1pb)		
Si $rra \geq rrb \rightarrow \text{coste} = rra (+1pb)$		Si $rrb \geq rra \rightarrow \text{coste} = rrb (+1pb)$

Mover el 3 arriba con $ra \rightarrow ra = 0$		Mover el 2 arriba con $rrb \rightarrow rrb = 3$
Sumamos $ra + rrb (3 + 1pb)$		

Mover el 3 arriba con $rra \rightarrow rra = 0$		Mover el 2 arriba con $rb \rightarrow rb = 2$
Sumamos $rra + rb (2 + 1pb)$		



## Del Stack a al Stack b



Buscamos en el stack a el número que tenga un coste directamente menor que el nodo del stack b de la parte superior (stack\_b->cost + 1; i++). Hacemos ra si está en half 0, rra si está en half 1. Cuando el stack b esté vacío, hacemos rra hasta que cost=1 esté en la parte superior.

## Del Stack b al Stack a