

Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación



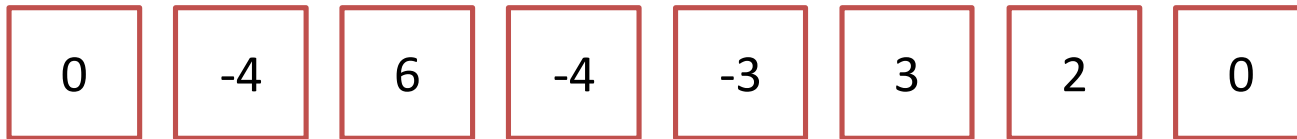
# IIC2115 – Programación como herramienta para la Ingeniería

Ayudantía 1: Estructuras de datos y Algoritmos

**Ayudante:** Pablo Seisdedos (pcseisdedos@uc.cl)

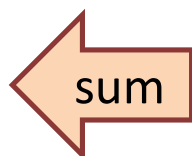
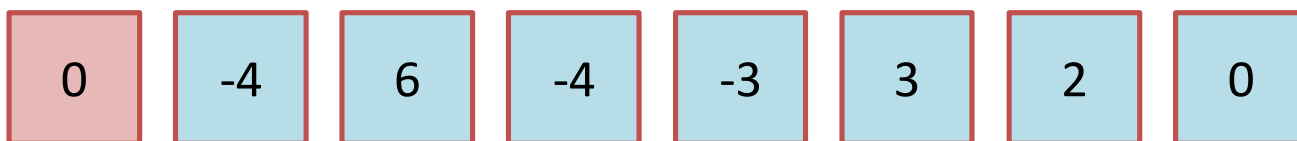
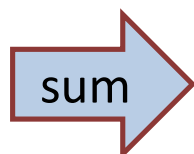
# Problema 1

Encontrar todos los índices tales que el lado izquierdo sume lo mismo que el lado derecho



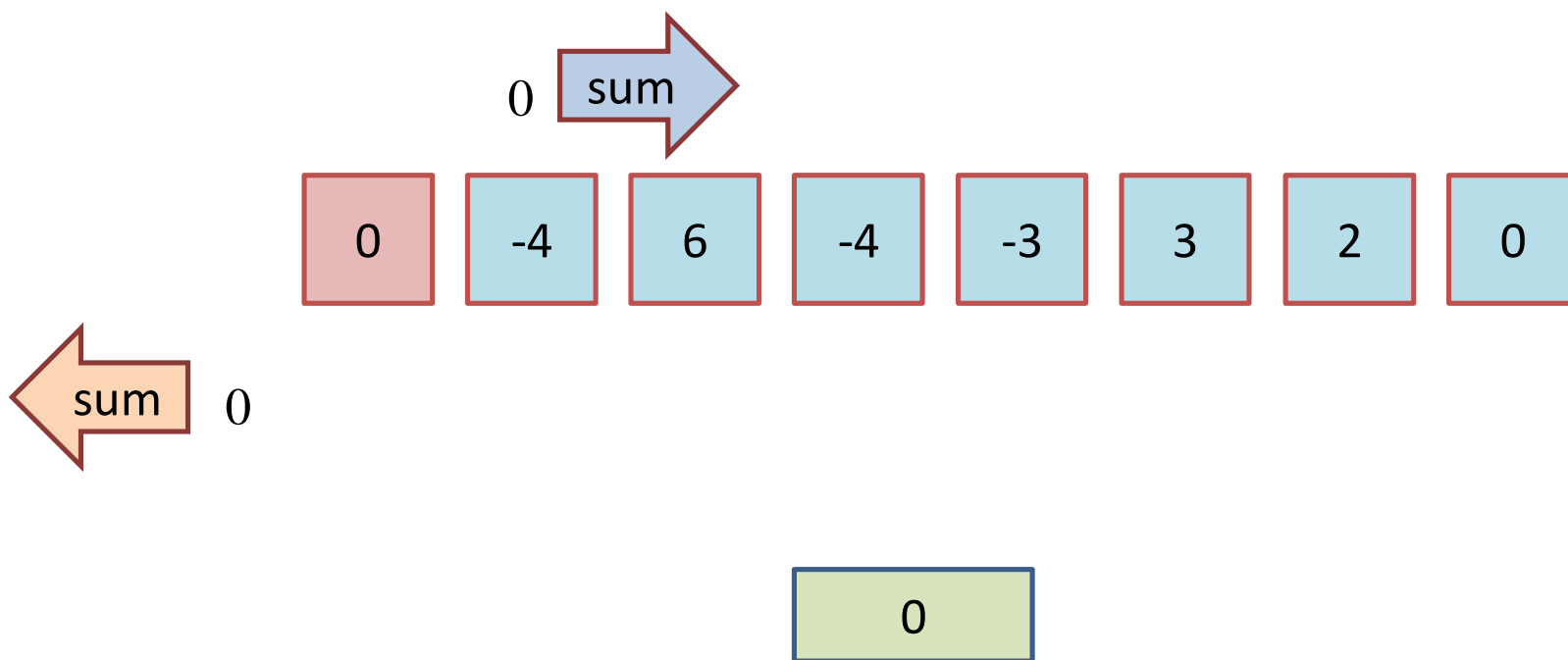
# Problema 1

$i = 0$ :



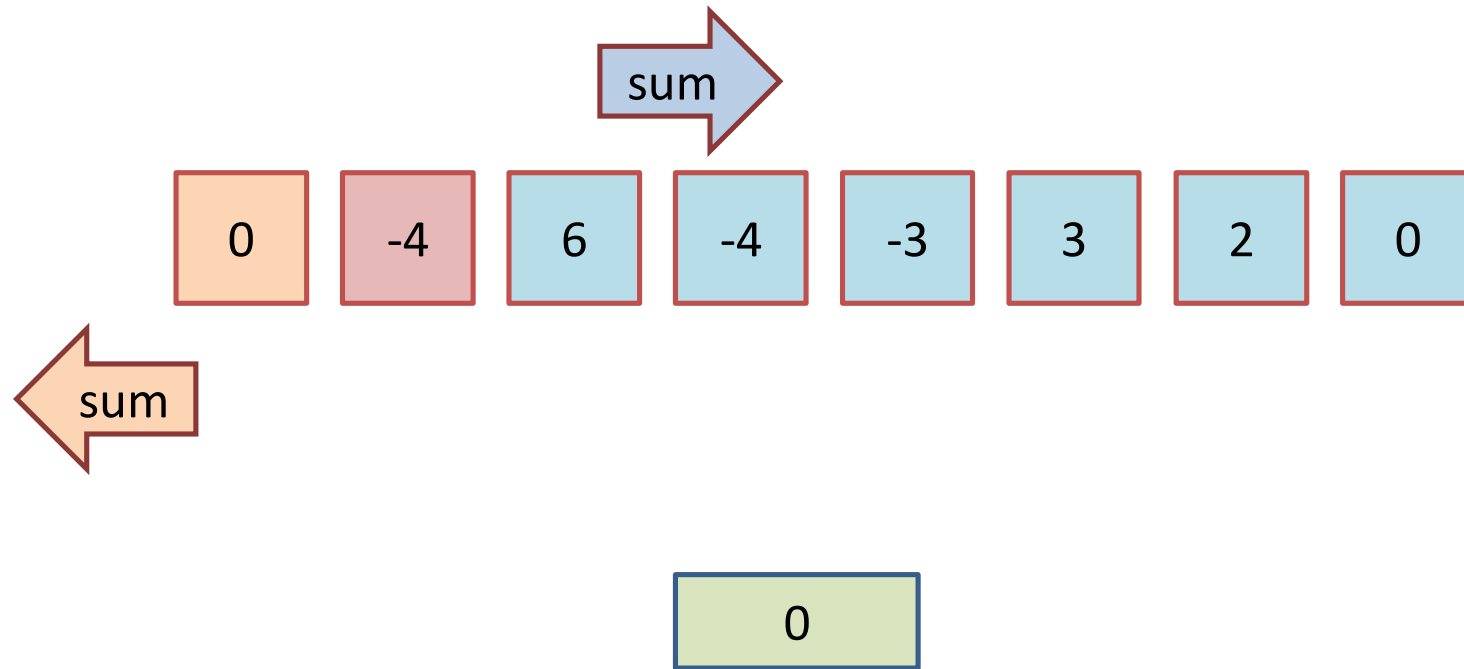
# Problema 1

$i = 0$ :



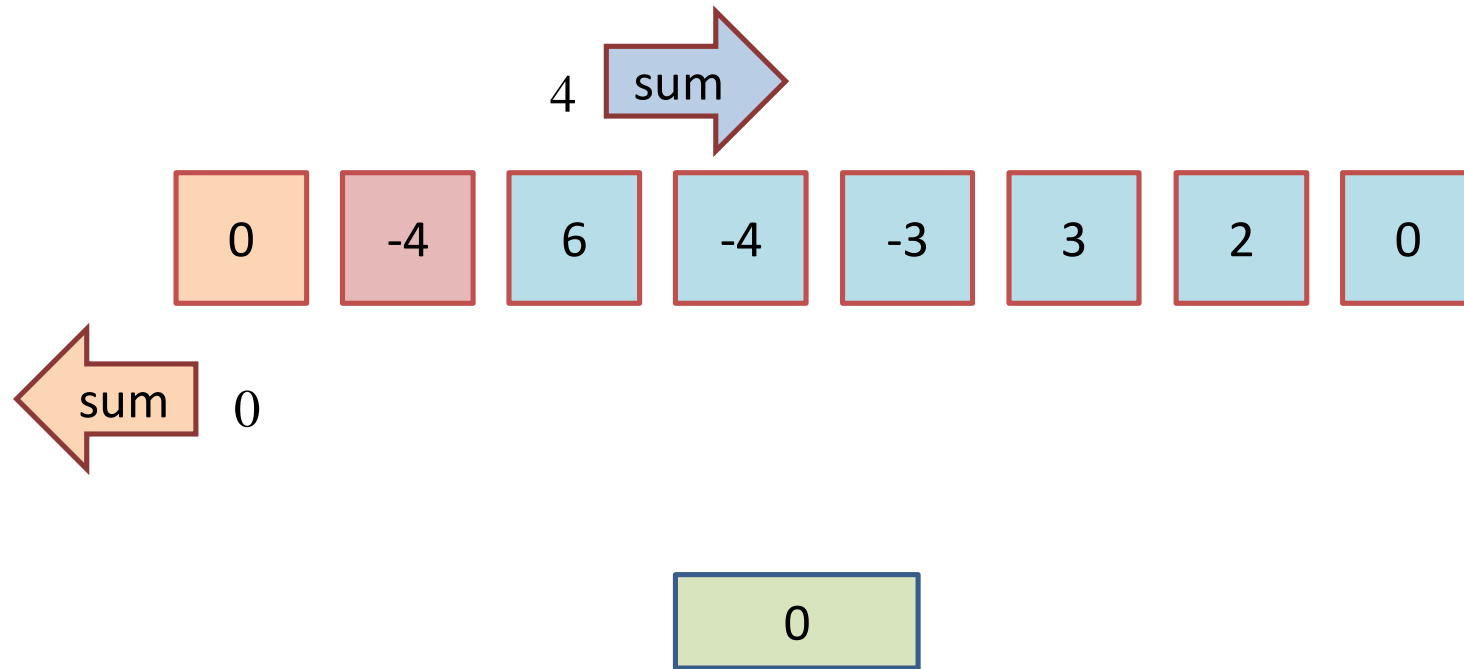
# Problema 1

$i = 1$ :



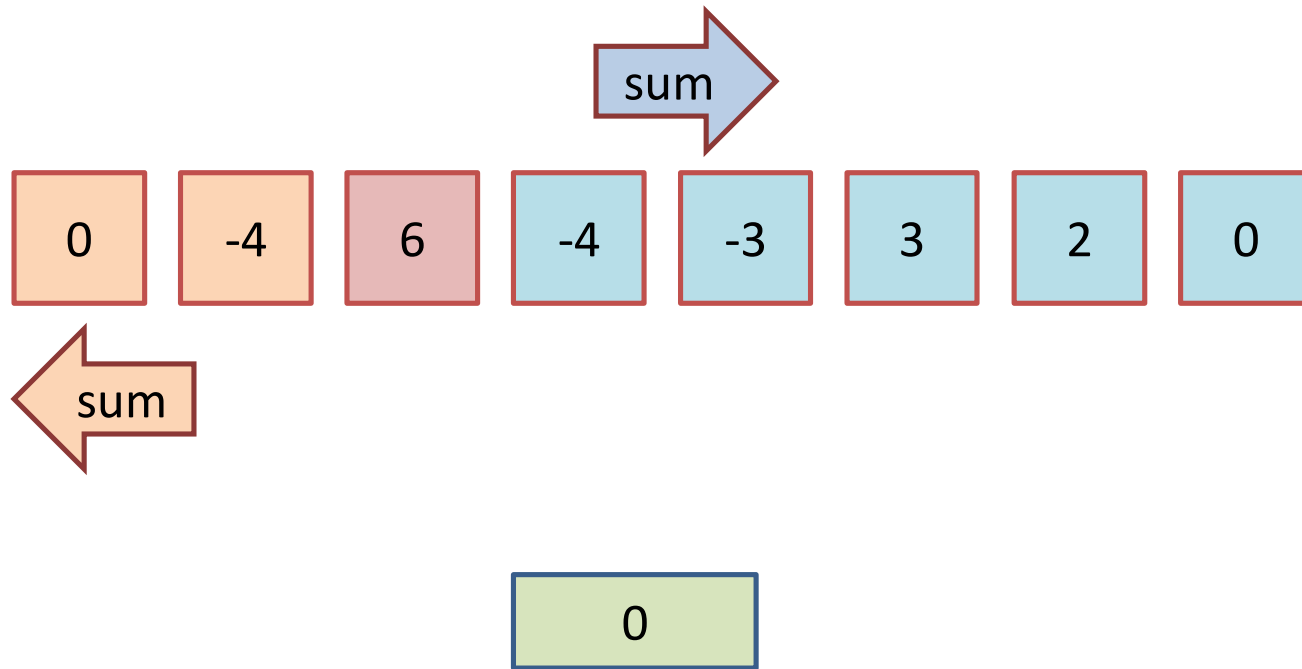
# Problema 1

i = 1:



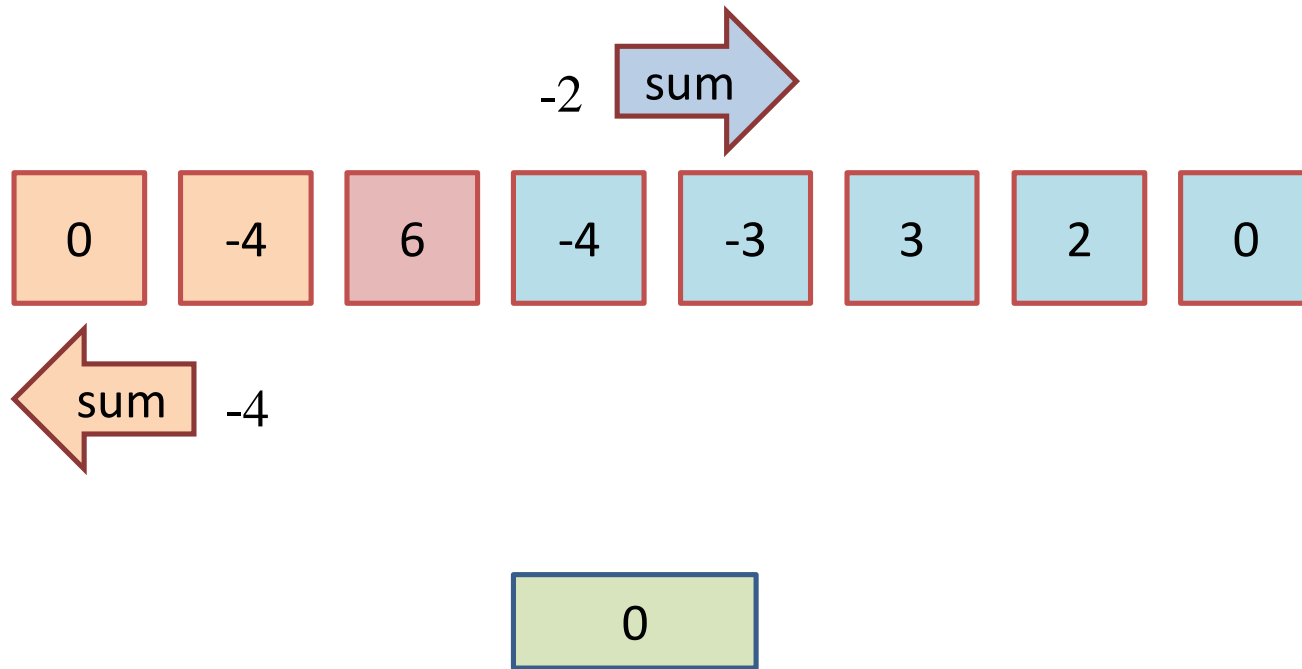
# Problema 1

$i = 2$ :



# Problema 1

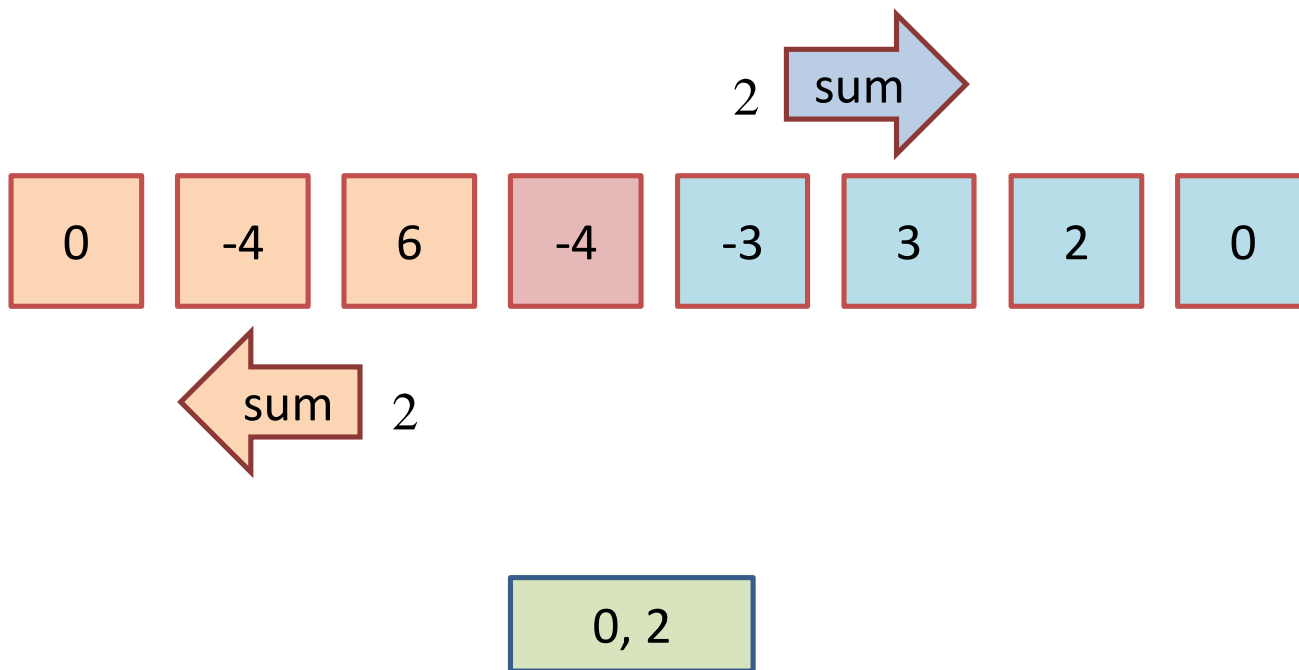
$i = 2$ :





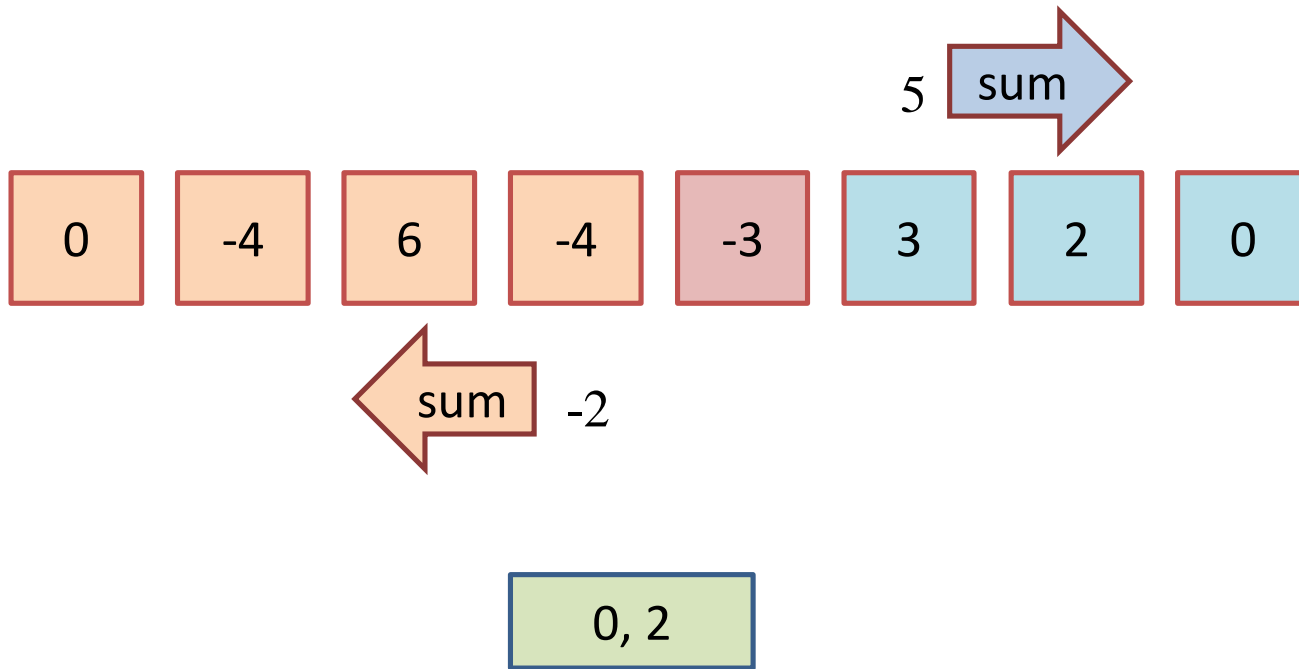
# Problema 1

$i = 3$ :



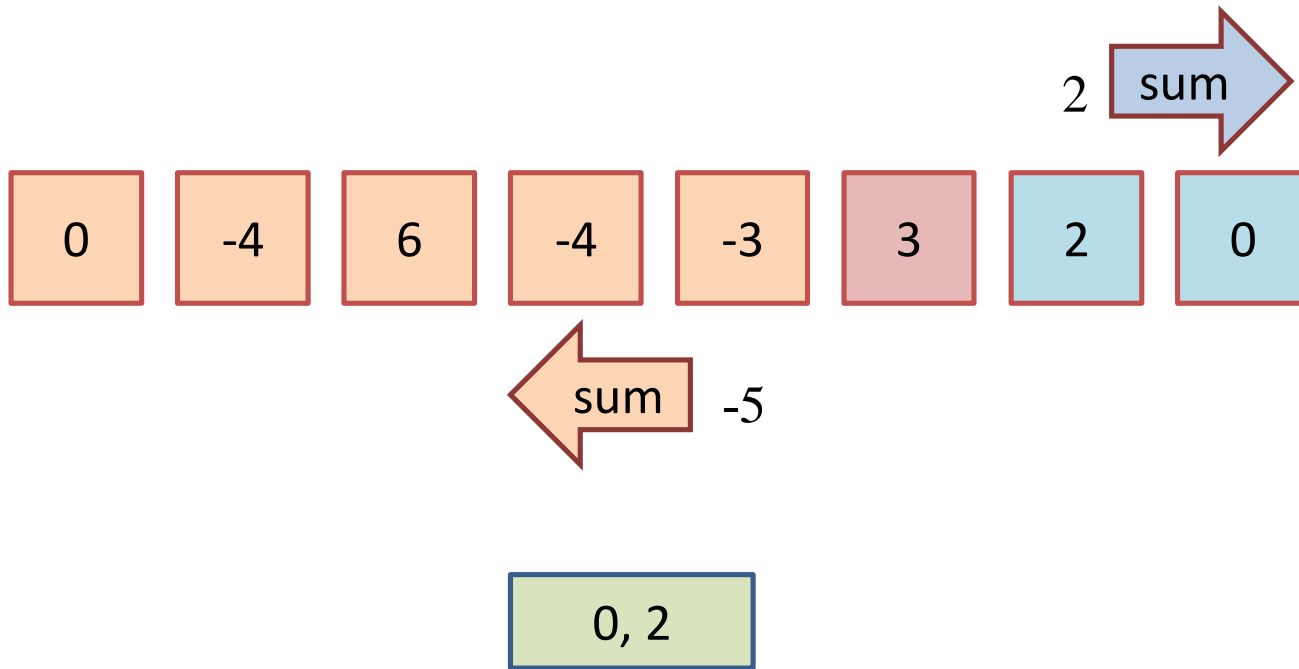
# Problema 1

$i = 4$ :



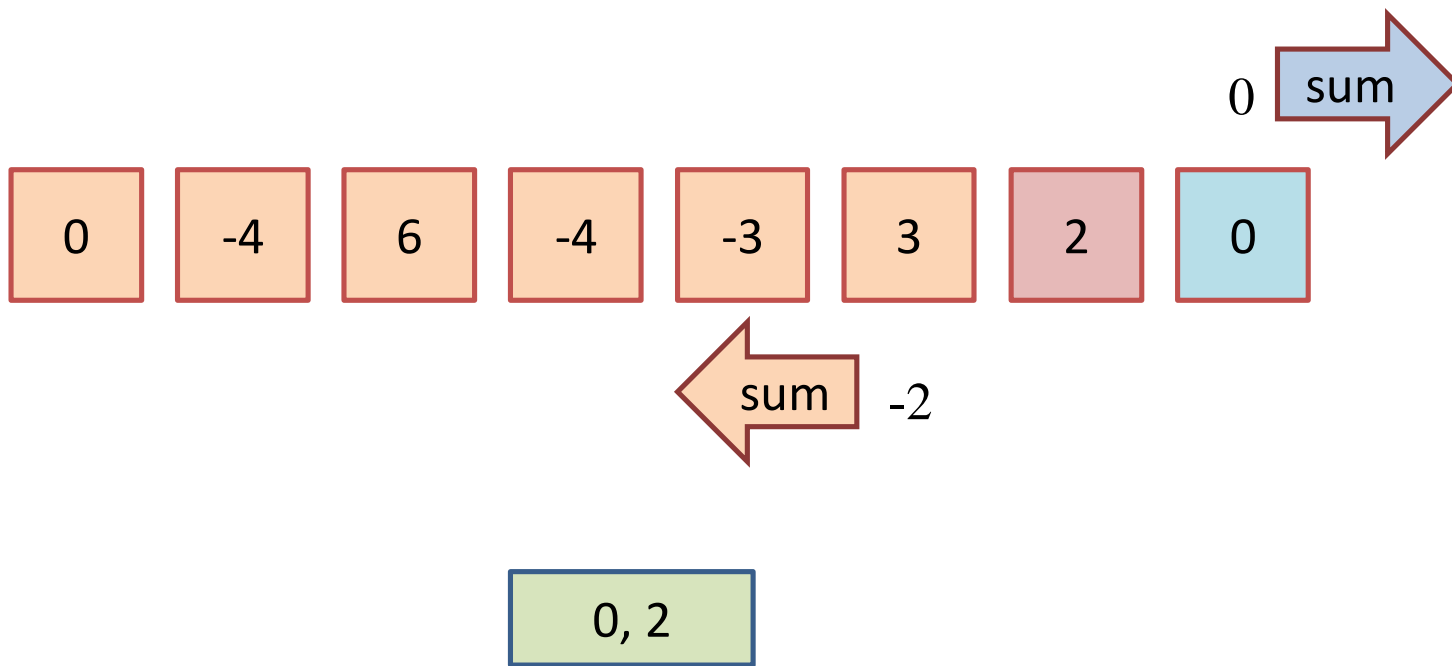
# Problema 1

$i = 5$ :



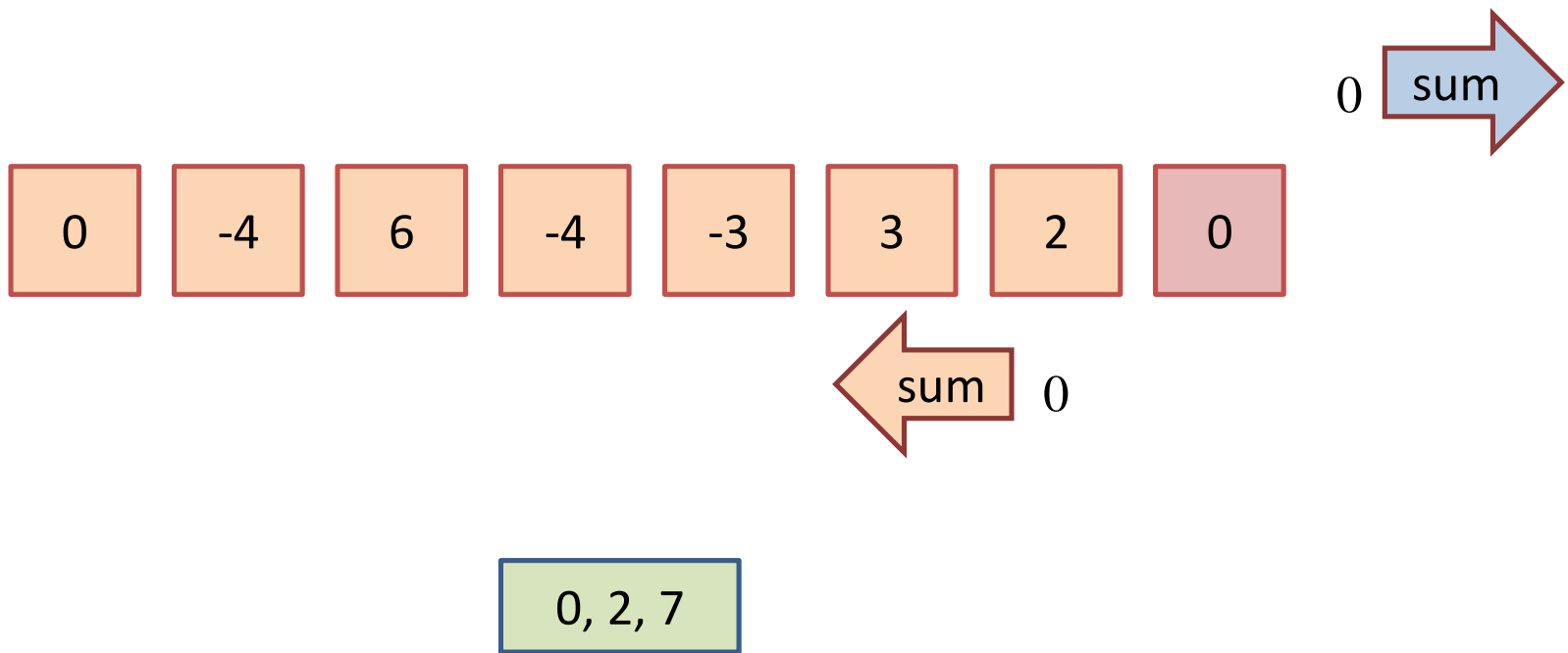
# Problema 1

i = 6:



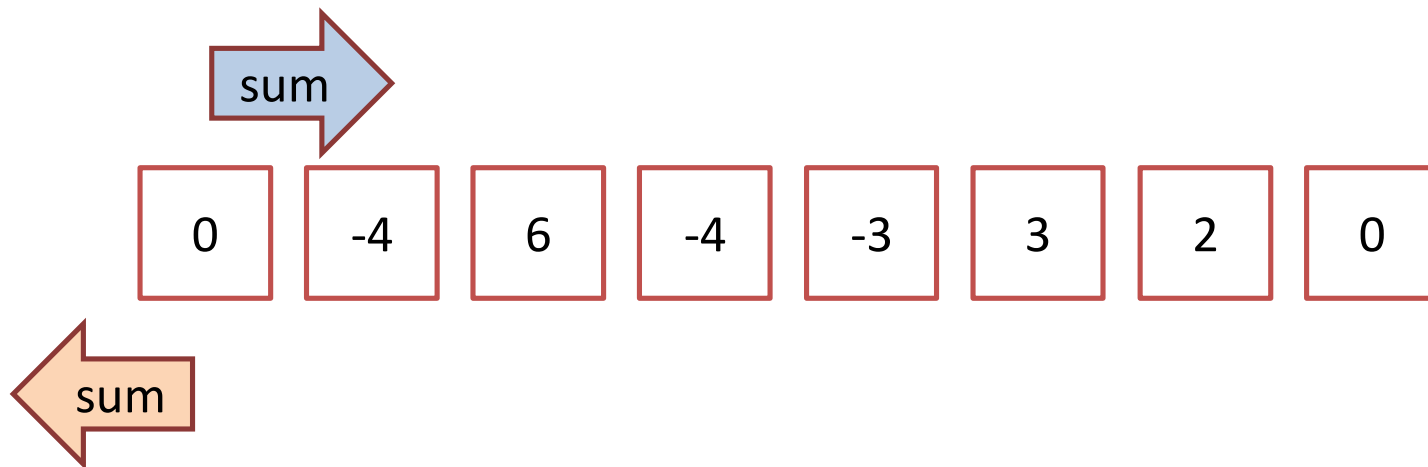
# Problema 1

$i = 7$ :



# Problema 1

Para cada  $i$ :



¿Cuál es la ineficiencia?

# Problema 2

Considere un conjunto de  $a$  alumnos ( $a \in \mathbb{N}^+$ ), donde cada uno tiene asociado una nota  $n$  ( $n \in \mathbb{N}^+$ ).

Escriba un programa que dado el conjunto de alumnos y sus notas, y un número  $k$  ( $k \in \mathbb{N}^+$ ), retorne  $k$  grupos de estudio disjuntos, tales que la suma de las notas parciales de los alumnos en cada grupo sea la misma.

## Backtracking: búsqueda con recursión eficiente

- Permite evitar recorrido de todo el árbol de recursión para encontrar una solución.
- Se almacena el último estado válido antes de la recursión, y se recupera si el nuevo estado no lo es.
- También puede entenderse como un stack, pero en este caso, al llevar registro del estado del problema, podemos evitar agregar más cosas al tope de este.



¿Qué es lo primero que debemos hacer?

Siempre lo primero es aterrizar el problema

- Dadas **N** notas, se buscan **K** grupos que sumen lo mismo

Siempre lo primero es aterrizar el problema

- Dadas **5** notas [8, 6, 2, 8, 4], se buscan **2** grupos que sumen lo mismo.
  - La suma total es  $8 + 6 + 2 + 8 + 4 = 28$
  - Como son 2 grupos, cada uno debe sumar 14
  - Ahora buscamos números que sumen 14: (8+6) y (8+2+4)
  - Por lo que, dados los *input*  $N=[8, 6, 2, 8, 4]$  y  $K=2 \rightarrow [[6,8], [2,4,8]]$

**¿Qué más es posible de concluir?**

Luego, extraemos reglas más generales

si  $\text{sum}(N) \% K \neq 0 \rightarrow$  Nunca se puede

si  $|N| < K \rightarrow$  Nunca se puede

$\forall n \in N, n \leq \text{sum}(N)/K$

si  $n = \text{sum}(N)/K$ ,  $n$  forma un grupo solo

A continuación planteamos el problema como un algoritmo

1. Verificar las condiciones que ya definimos.
2. Si existen alumnos con nota igual a  $\frac{\text{sum}(N)}{K}$ :
  - a) Extraerlos de la lista
  - b) Disminuir el valor de K (grupos que faltan)
  - c) Agregar un grupo de un alumno
3. ¿Cómo iteramos?

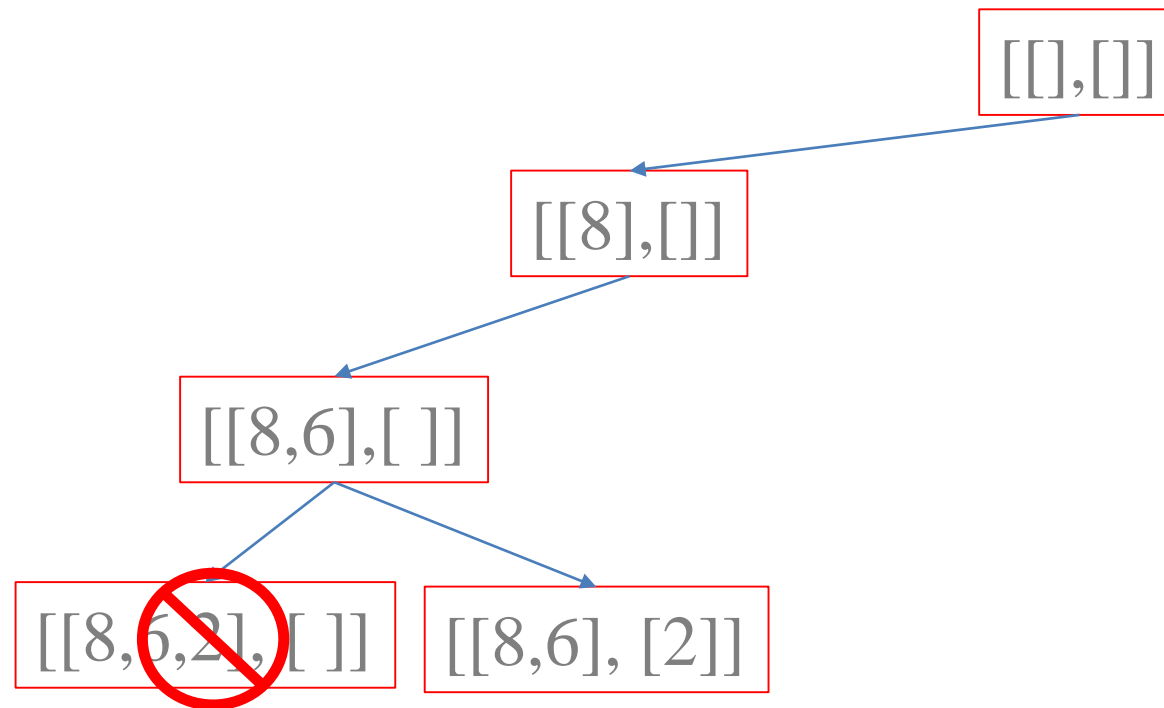
(Casi) Finalmente, seguimos el algoritmo

$n = [8, 6, 2, 8, 4]$		$\text{notas} = 5$
$k = 2$	$\rightarrow$	$\text{suma} = 28$
		$\text{suma\_por\_grupo} = 28/2 = 14$

- El número de notas (5) es mayor al valor de  $k$  (2)
- La suma total (28) es dividida perfectamente por  $k$  (2) (no deja resto)
- No hay elementos mayores que  $\text{suma\_por\_grupo}$  (14)

- Tenemos que formar dos grupos

$[8, 6, 2, 8, 4]$



¿Cuándo paro?

*¿Qué reglas podemos definir para terminar de iterar?*

- Si algún grupo  $>$  que el objetivo  $\rightarrow$  No hay solución en esa rama
- Si asigné todas las notas  $\rightarrow$  Encontramos solución



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación



# IIC2115 – Programación como herramienta para la Ingeniería

Ayudantía 1: Estructuras de datos y Algoritmos

**Ayudante:** Pablo Seisdedos (pcseisdedos@uc.cl)