

## Problema de la suma de subconjuntos

"dado un conjunto de enteros, ¿existe algún subconjunto cuya suma sea exactamente cero?"

### Análisis

Teniendo un array de  $n$  enteros se debe calcular si al sumar algún subconjunto el resultado es 0

### Requerimientos funcionales

1. Pedir al usuario el tamaño del array
2. El usuario llenará el array
3. Obtener subconjuntos y hacer su suma
4. Indicar al inicio el total de subconjuntos
5. Indicar los subconjuntos cuya suma fue 0

### Requerimientos no funcionales

6. Mostrar en consola todos los subconjuntos y su suma enumerados
7. Almacenar una lista con los números de los subconjuntos cuando sumen 0

### Diseño

- Se realizará en java usando netbeans
- Se hará uso de estructuras de control (if, while, for, do while, if-else)
- Realizarlo en programación estructurada
- Al finalizarlo se evaluará si puede pasarse a programación paralela
- Uso de try-catch

## Especificaciones de los requerimientos

- ①. El tamaño será guardado en una variable entera  $n$  (global)
  - Será solicitado con un JOptionPane y verificar que sea entero  $> 2$
  - No se puede avanzar hasta que  $n \neq 0$
- ②. Se usará JOptionPane y verificar que sean enteros
  - El int[] se llamará conjunto (global)
- ④. La fórmula que se usará es:

$$\sum_{r=2}^n \left( \frac{n!}{(n-r)! \cdot r!} \right)_r = \text{total subconjuntos}$$

- ③. Para obtener los subconjuntos no se tomarán en cuenta las permutaciones, solo combinaciones
- ⑥. Se tendrá un contador (cont) para enumerar los conjuntos encontrados
- ③. Sumar los elementos del subconjunto mediante un for.
- ⑦. Si la suma del subconjunto es  $= 0$  se almacena en un ArrayList (respuesta) el número del subconjunto (cont).
- ⑤. Verificar el size() de respuesta e informar el resultado