

Ejercicio Backtracking

Clase #27

IIC1103 – Introducción a la Programación

Marcos Sepúlveda (marcos@ing.puc.cl)

Subconjunto posible

- ▶ Ver si me alcanza con las monedas que tengo para pagar un monto exacto
- ▶ Ver si llenando un subconjunto de autos a su máxima capacidad, puedo llevar a un grupo de amigos a una fiesta



Subconjunto posible

- ▶ Dada una lista de enteros positivos (**numeros**) y un valor entero deseado (**objetivo**), ¿hay un subconjunto de elementos de la lista que suma el valor objetivo?
- ▶ Podría haber más de un subconjunto. Basta que muestre uno cualquiera.
- ▶ Por ejemplo, si **numeros**=[8, 6, 7, 5, 3, 10, 9] y **objetivo**=15, cualquiera de los subconjuntos [8, 7], [6, 9], [7, 5, 3] o [5, 10] suma 15.
- ▶ Por otro lado, si **numeros**=[11, 6, 5, 1, 7, 13, 12] y **objetivo**=15, ningún subconjunto suma 15.

Subconjunto posible

- ▶ En concreto, se te pide que implementes –usando backtracking– la función **`resolver(numeros, objetivo)`**, que retorne una lista de enteros con **UNA** posible solución. Si no hay solución, debe retornar una lista vacía.
- ▶ La función debe recibir y retornar los parámetros indicados, pero puedes definir todas las funciones adicionales que quieras (puedes implementar el backtracking en otra función).