



IIC2115 – Programación como Herramienta para la Ingeniería (I/2020)

Taller de revisión de ejercicios - Técnicas de programación

Problema 1

Dada una lista de números enteros (tanto positivos, como negativos), encuentre todos los elementos de la lista que realizan una partición de esta en dos grupos de igual suma. Se pide retornar el índice de los números que cumplen la condición. Por ejemplo, si la lista fuese [3,4,2,1] la respuesta debiese ser [1], dado que el el número 4 divide a la lista en [3] y [2,1] que suman lo mismo.

Código:

```
def division_lista(lista):  
    # Solucion del problema  
  
    lista = [0,-4,6,-4,-3,3,2,0]  
    indices = division_lista(lista)  
    print(indices)
```

Salida:

[0,3,7]

- (a) Resuelva el problema utilizando completamente fuerza bruta, es decir, analice todos los casos para entregar su respuesta y no use ninguna lógica que le ahorre tiempo de resolución. ¿De qué orden es la

solución construida?

- (b) Resuelva el mismo problema pero ahora debe reducir el orden de solución. Es decir, trate de optimizar el tiempo de solución de la parte (a). ¿De qué orden es la solución construida?
- (c) Construya dos gráficos *tamaño de input vs tiempo de ejecución* para las soluciones de la parte (a) y (b). ¿qué observa?

Problema 2

Escriba un programa que cuente las formas en que se puede obtener un número objetivo entero mayor que cero a partir de elementos pertenecientes a una lista de números (sin repetirlos). Para ello, solo puede utilizar números de adición (+) o substracción (-). Por ejemplo, la cantidad de veces que se puede formar el número 6 con la lista [5, 3, -6, 2] es 4. Estas serían:

1. $-(-6) = 6$
2. $+(5)+(3)-(2) = 6$
3. $+(5)-(3)-(-6)-(2) = 6$
4. $-(5)+(3)-(-6)+(2) = 6$

Código:

```
def numeros(lista, objetivo):  
    # Solucion del problema  
  
    lista = [8, -2, 1, -3, 5]  
    objetivo = 5  
    conteo = numeros(lista, objetivo)  
    print(conteo)
```

Salida:

7