

3.º parcialito – 29/11/2021

Resolvé los siguientes problemas en forma clara y legible. Podés incluir tantas funciones auxiliares como creas necesarias.

Debés resolver 3 de los siguientes ejercicios. Los ejercicios a resolver están definidos por tu padrón. Podés revisar en [esta planilla](#).

1. Dado un arreglo de enteros y un número  $K$ , se desea que todos los elementos del arreglo sean mayores a  $K$ . Aquellos números que sean menores o iguales a  $K$  deberían combinarse de la siguiente forma: buscar los dos números más chicos del vector, sacarlos y generar uno nuevo de la forma “Nuevo número = número-más-chico + 2 x segundo-más-chico”. Por ejemplo, si  $K = 10$  y los números más chicos del arreglo son 3 y 4:  $3 + 2 * 4 = 11$ . Los números combinados pueden volver a ser combinados con otros en caso de ser necesario (en el ejemplo no lo es), aplicando la misma lógica hasta que el número resultante sea mayor a  $K$ .

Implementar una función que reciba un arreglo de enteros, su largo y un número  $K$ , y devuelva una lista con los elementos que quedarían luego de aplicar las modificaciones. El arreglo original debe quedar en el estado original. El orden de la lista no es importante. En caso de no poderse combinar todos los elementos para que todos los elementos sea mayores a  $K$ , devolver NULL. Determinar y justificar la complejidad del algoritmo implementado.

Ejemplo: Entrada: [11, 14, 8, 19, 42, 3, 1, 9];  $K = 13$ :

```
[11, 14, 8, 19, 42, 3, 1, 9] - (1,3) -> 1 + 2*3 = 7
[11, 14, 8, 19, 42, 9, 7]    - (7,8) -> 7 + 8*2 = 23
[11, 14, 19, 42, 9, 23]      - (9,11) -> 9 + 11*2 = 31
[14, 19, 42, 23, 31]
```

Notar que si el 9 no estuviera en el arreglo, no se podría resolver el problema (debemos devolver NULL), ya que el 11 no podría combinarse con ningún otro número.

2. Dado un arreglo de enteros y un número  $K$ , se desearía que todos los elementos del arreglo sean menores o iguales a  $K$ . Aquellos números que sean mayores a  $K$  deberían combinarse de la siguiente forma: buscar los dos números más grandes del vector, sacarlos y generar uno nuevo de la forma “Nuevo número = 2 x segundo-más-grande - número-más-grande”. Por ejemplo, si  $K = 10$  y los números más grandes del arreglo son 15 y 12:  $2 * 12 - 15 = 9$ . Los números combinados pueden volver a ser combinados con otros en caso de ser necesario (en el ejemplo no lo es), aplicando la misma lógica hasta que el número resultante sea menor o igual a  $K$ .

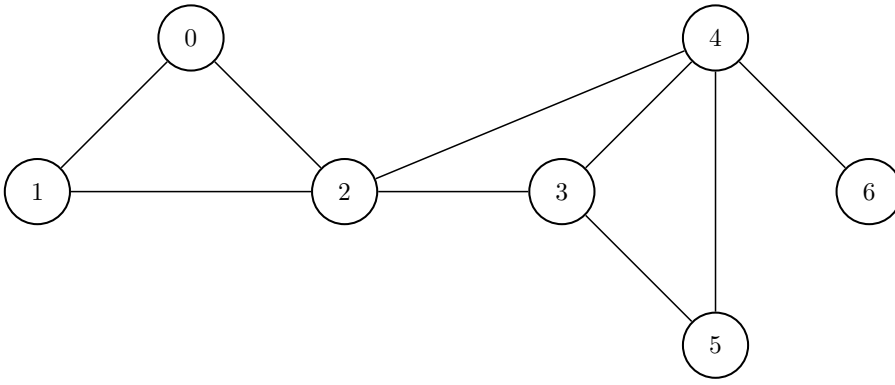
Implementar una función que reciba un arreglo de enteros, su largo y un número  $K$ , y devuelva la cantidad de combinaciones que deberían ser necesarias para que el arreglo tenga todos los números menores o iguales a  $K$  como se indica. El arreglo original debe quedar en el estado original. En caso de no poderse combinar todos los elementos para que todos los elementos sean menores o iguales a  $K$ , devolver -1. Determinar y justificar la complejidad del algoritmo implementado. Aclaración: la función sólo debe devolver cuántas combinaciones se deberían hacer, no debe devolver cómo quedarían los elementos al final.

Ejemplo: Entrada: [11, 14, 8, 19, 42, 12, 5, 1, 9];  $K = 9$ :

```
[11, 14, 8, 19, 42, 12, 5, 1, 9] - (19, 42) -> 2 * 19 - 42 = -4
[11, 14, 8, 12, 5, 1, 9, -4]    - (12, 14) -> 2 * 12 - 14 = 10
[11, 8, 5, 1, 9, -4, 10]        - (10, 11) -> 10 * 2 - 11 = 9
[8, 5, 1, 9, -4, 9]
```

Notar que si  $K = 8$ , no se podría resolver el problema (debemos devolver -1). Esto es porque los dos 9s se combinarían para formar un nuevo 9, que luego no se podría combinar con otro número.

3. Implementar una **función** que reciba un grafo no dirigido y pesado (implementado con matriz de adyacencia) y devuelva una lista que sea equivalente a la representación de lista de adyacencia del mismo grafo (implementada como lista de listas). Indicar y justificar el orden del algoritmo implementado.
4. Implementar una **función** que reciba un grafo dirigido y no pesado (implementado con listas de adyacencia (diccionario de diccionarios)) y devuelva una matriz que sea equivalente a la representación de matriz de incidencia del mismo grafo. Indicar y justificar el orden del algoritmo implementado.
5. Hacer un seguimiento sobre el siguiente grafo para obtener los puntos de articulación del mismo. Explicar qué estructuras auxiliares se utilizan, y cómo (sobre el mismo seguimiento). Si pudieras agregar una arista en el grafo para que no presente puntos de articulación, ¿cuál sería? Explicar brevemente por qué esa arista logra lo indicado.



6. Hacer un seguimiento sobre el siguiente grafo para obtener las componentes fuertemente conexas del mismo. Explicar qué estructuras auxiliares se utilizan, y cómo (sobre el mismo seguimiento). Si pudieras agregar una arista en el grafo para que haya una única componente fuertemente conexas, ¿cuál sería? Explicar brevemente por qué esa arista logra lo indicado.

