

# Analisis de algoritmos 2022-1

Tarea 9. Primera tarea opcional, permite obtener puntos extra.

Fecha de entrega: miércoles 19 de enero de 2022

## 1 Ejercicios

Cada uno de los siguientes problemas permite obtener hasta un 0.4 de calificación extra; con un total de a lo más, 0.8 puntos.

- Propón un algoritmo de tiempo  $O(n \log n)$  que, dado un arreglo de  $n$  elementos comparables entre sí, con potenciales repeticiones de sus valores; devuelva un arreglo con los mismos valores que la entrada, pero sin repeticiones, y tal que el orden de los elementos de la salida sea el que corresponda a las primeras ocurrencias de cada valor original.

Instancias de ejemplo:

- Entrada:  $[3, 5, 2, 1, 3, 5, 4, 6, 1]$ , salida:  $[3, 5, 2, 1, 4, 6]$  .
- Entrada:  $[2, 1, 2, 1, 2, 1, 3]$ , salida:  $[2, 1, 3]$  .

Tip: Considera procesar elementos de la forma:  $(valor, índice)$ . Estas tuplas de dos elementos te permitirán registrar la posición del arreglo original en donde se encontraba el valor *valor*, que te permitirá calcular las posiciones en donde deben ir los valores que contiene el arreglo de salida.

- Considera un arreglo infinito  $A$  cuyas primeras  $n$  celdas contienen enteros en orden creciente, y el resto de las celdas tienen el símbolo  $\infty$ . No conoces el valor de  $n$ . Describe un algoritmo que tome un entero  $x$  como entrada y devuelva una posición en el arreglo  $A$  que contenga el valor  $x$ , si semejante posición existe, o -1 si no existe, en tiempo  $O(\log n)$ . (Si te inquieta el hecho de que el arreglo  $A$  sea infinito, puedes pensar que su longitud es  $n$ , pero no conoces el valor de  $n$ , y que la implementación del tipo de datos arreglo te devolverá el símbolo  $\infty$  cuando accedas a una posición más allá de la  $n$ -ésima).

## 2 Rúbrica de evaluación

Consideren los siguientes puntos a revisar para la elaboración de su tarea.

1. Para todos los ejercicios tener presente:
  - El algoritmo está bien planteado.
  - La prueba contempla todos los detalles.
  - La prueba está libre de detalles innecesarios.
  - El lenguaje utilizado es simple y claro.
2. Para las partes secuenciales de los algoritmos que requieren el uso de invariante tener presente:
  - El invariante está bien planteado.
  - El argumento del porqué el invariante implica la corrección del algoritmo es correcto y completo.
3. Para las partes recursivas de los algoritmos que requieren el uso inducción matemática tener presente:
  - El caso base está bien planteado.
  - La hipótesis de inducción está bien planteada.
  - El caso inductivo está bien planteado.
  - El caso inductivo hace uso explícito de la H.I.
4. Argumenta con cuidado el porqué el algoritmo tiene la complejidad temporal solicitada.