



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2133 — Estructuras de Datos y Algoritmos
2021 - 2

Ayudantía 2

Pregunta 1

La función *Merge* recibe como parámetro 2 secuencias ordenadas A y B y las combina en una sola secuencia ordenada que contiene los datos de ambas secuencias A y B. Podemos asumir que *Merge* es correcto y toma tiempo $\mathcal{O}(n)$.

El algoritmo de ordenación *MergeSort*, divide un arreglo en mitades recursivamente hasta que obtiene subarreglos de largo 1. Luego los ordena utilizando *Merge*, hasta terminar con el arreglo original ordenado.

- a) Demuestre que *MergeSort* es correcto.
- b) Calcule su complejidad.

Pregunta 2

El algoritmo de ordenamiento QuickSort no se comporta bien cuando existen muchos datos repetidos en el arreglo a ordenar.

- a) ¿Por qué sucede esto?
- b) Proponga una modificación al algoritmo visto en clase que permita mejorar esta situación.

Propuesto - Pregunta I1 2021-1

A un alumno que estaba estudiando QuickSort se le ocurrió que podría usar el algoritmo median visto en clases para mejorar QuickSort. Simplemente modifica median para que retorne el índice de la mediana (en vez del valor de la mediana), y reemplaza la llamada a partition por una llamada a median modificado en la definición de QuickSort. Así, no tendríamos que arriesgarnos a elegir malos pivotes, pues ya sabemos encontrar la mediana y podemos lograr siempre el mejor caso de QuickSort.

Nota: asume que no hay elementos repetidos.

- a) ¿Cuál es la complejidad de la idea de tu compañero en el mejor caso?
- b) ¿Cuál es la complejidad de la idea de tu compañero en el peor caso?
- c) ¿Qué le dirías a tu compañero al respecto de si su idea es una mejora a QuickSort? Argumenta.