



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

IIC2133 — Estructuras de Datos y Algoritmos
2021 - 1

Pauta Tarea 1

Component Tree vs Max Tree (50%)

- Eficiencia de memoria: Como se menciona en enunciado, cada nodo del *Component Tree* guarda todos los pixeles que tienen sus descendientes, a diferencia del *Max Tree*, que en sus nodos guarda los pixeles de un cierto valor específico. De esto, se desprende que un *Component Tree* utiliza más memoria que un *Max Tree*, al ser este último una representación compacta del primero. Notar que hacer una distinción entre eficiencias de memoria es importante, ya que en la práctica ambos árboles permiten las mismas acciones. (3 **puntos**).
- Ventajas/usos del Component Tree: Permite acceder (y modificar) los valores de los pixeles de un vecindario directamente sobre un nodo, en lugar de tener que descender totalmente en el árbol. Se aceptan otras ventajas, mientras estén correctamente justificadas (1.5 **puntos**).
- Ventajas/usos del *Max Tree*: Permite implementar filtros sobre valores específicos de pixeles de manera más simple. En el *Component Tree* se requiere ir revisando los valores de cada pixel de un nodo. Se aceptan otras ventajas, mientras estén correctamente justificadas (1.5 **puntos**).

Complejidad (50%)

Sea n el número de pixeles, y d la profundidad del árbol.

- Creación del arbol: Depende de la implementación. Hay implementaciones en $O(n)$. La clásica es $O(nd)$, en los que para cada profundidad se revisan los pixeles restantes. Si para hacer los vecindarios revisa todos los otros pixeles, en lugar de solo los vecinos, entonces la implementación puede ser $O(n^2d)$ (3 **puntos**).
- Aplicación de filtros: Aplicar el filtro requiere revisar cada nodo del arbol una sola vez. En el peor caso, todos los nodos tienen un solo pixel (no pueden haber nodos vacíos), por lo que la complejidad del filtro es $O(n)$. Las complejidades pueden ser distintas, dependiendo de la implementación. (1.5 **puntos por filtro**).