



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

IIC2133 — Estructuras de Datos y Algoritmos  
2020 - 2

## Pauta Tarea 1

### Construcción del árbol

La construcción de kd-tree depende del método que se use para conseguir la mediana:

1. Caso quickselect:

- **Caso promedio:  $O(S \log S)$ :** Quickselect es un algoritmo en promedio  $O(S)$ , que se realiza  $O(\log S)$  veces. Por lo tanto,  $O(S \log S)$ .
- **Peor caso:  $O(S^2)$ :** Quickselect es  $S^2$  en el peor caso. Por lo tanto,  $O(S^2)$ .

2. Usando ordenación  $O(S \log S)$  (quicksort, etc):

- **Caso promedio:  $O(S \log^2 S)$ :** Se realiza el algoritmo  $O(S \log S)$  un número  $O(\log S)$  de veces. Por lo tanto,  $O(S \log^2 S)$ .
- **Peor caso:  $O(S^2)$ :** Depende del algoritmo que se use para ordenar, pero tomando como ejemplo quicksort, para el cual es  $O(S^2)$  en el peor caso. En la instancia de que el peor caso del algoritmo de ordenación esté acotado por  $O(S \log S)$ , la complejidad final será  $O(S \log^2 S)$ .

→ Si la complejidad en el peor caso es errónea ( $O(S^2 \log S)$ ): (0.5 pts) por complejidad y (0.5pts) pts por justificación.

### Búsqueda en el árbol

Cualquier caso  $O(\log S)$ . Justificar que en cada nivel del árbol se separa la muestra en dos, por lo que la altura de este está acotada por  $O(\log S)$ , esto se debe a que la mediana genera un árbol balanceado.

### Programa completo

Depende de la construcción. Se ejemplifica caso promedio de quickselect:

$$O(\text{free}) = O(S) \text{ (todos los nodos del árbol)} + O(P)$$

$$O(\text{Programa}) = O(\text{construcción}) + O(\text{búsqueda} * F * P) + O(\text{free})$$

$$O(\text{Programa}) = O(S \log S) + O(F P \log S) + O(S) + O(1)$$

Considerando que  $S < FP$  (suposición razonable dados los tests), se obtiene:

$O(\text{Programa}) = O(F \log S)$ .

→ Si se muestra solo  $O(\text{Programa}) = O(\text{búsqueda} * F * P)$ , es decir, no se muestra la complejidad de construcción y de liberar. Entonces: (0.75 pts) por complejidad y (0.375 pts) por justificación.

→ Si es que muestra solo  $O(\text{Programa}) = O(\text{construcción}) + O(\text{búsqueda} * F * P)$ , es decir, no muestra la complejidad de liberar, y al momento de explicar el algoritmo sí menciona la liberación de memoria. Entonces: (0.75 pts) por complejidad y (0.55 pts) por justificación.

## Cálculo empírico

- (0.75 pts) por mostrar tablas y/o gráficos que denoten una mejoría sustancial.
- (0.75 pts) por indicar por qué es significativo y cómo afectan cada uno de los parámetros.

## Bonus

- **Bonus 1:** (0.5 pts) por profundizar en que un tamaño de hoja más pequeño hace que el programa sea más rápido, deben justificar con datos concretos.
- **Bonus 2:** (0.5 pts) se puede utilizar rango, varianza, desviación estándar, entropía, gini, ganancia de información, etc. Otra vez, se debe justificar con datos empíricos.

## Puntajes

- Para todos los cálculos de complejidad: 0.75 pts.
- Para todas las justificaciones de complejidad: 0.75 pts.
- Total por ítem: 1.5 pts.