

INTRODUCCIÓN

Quickselect es un algoritmo para seleccionar el k-ésimo elemento más pequeño de un arreglo desordenado. El algoritmo divide el arreglo en función de un pivote elegido al azar, de manera que todos los elementos menores al pivote se encuentren a su izquierda y los mayores a su derecha hasta encontrar el elemento que buscamos en subproblemas más pequeños.

RECURRENCIA

Caso promedio: El algoritmo selecciona aleatoriamente el pivote y divide el arreglo en dos subproblemas del mismo tamaño.

$$T(n) = (n/2) + O(n)$$

Peor caso: El pivote es el elemento más grande o más pequeño de la lista y solo va reduciendo en 1 el tamaño del arreglo.

$$T(n) = T(n-1) + O(n)$$

BUCLE INVARIANTE

Bucle invariante: Tenemos dos punteros L y R que apuntan al inicio y final del arreglo, el subarreglo $A[L...R]$ contiene elementos menores o iguales al pivote P a la izquierda de P, y elementos mayores o iguales a P a la derecha de P.

Esta invariante se cumple a lo largo del desarrollo de todo el algoritmo, siendo verdadera en su inicialización, mantenimiento y terminación.

```
function quickselect[A, k, L, R]:
  if L = R:
    return A[L]
  else:
    pivot_index = partition[A, L, R]
    pivot_rank = pivot_index - L + 1
    if k = pivot_rank:
      return A[pivot_index]
    elif k < pivot_rank:
      return quickselect[A, k, L, pivot_index - 1]
    else:
      return quickselect[A, k - pivot_rank, pivot_index + 1, R]
```

PSEUDOCÓDIGO

```
function partition[A, L, R]:
  Choose a pivot element p from A
  while L <= R:
    while A[L] < p:
      L = L + 1
    while A[R] > p:
      R = R - 1
    if L <= R:
      Swap A[L] and A[R]
      L = L + 1
      R = R - 1
  return L
```

ALGORITMO

Para implementar Quickselect necesitamos dos funciones:

- Partition: Divide el arreglo en 2 subarreglos más pequeños
- Quickselect: selecciona el subarreglo que contenga el k-ésimo elemento más pequeño que buscamos

COMPLEJIDAD

La complejidad del tiempo en el peor de los casos del algoritmo Quickselect es $O(n^2)$, ocurre cuando en todas las ocasiones que se divide el arreglo elegimos un mal pivote.

La complejidad del tiempo en el caso promedio de Quickselect es $O(n)$ cuando el elemento pivote se elige al azar y en promedio las particiones son balanceadas.

APLICACIÓN

Perfecto para listas desordenadas, cuyo cierto elemento más pequeño o más grande deseamos conocer. Por ejemplo: conocer el tercer platillo más caro de un restaurante, dada la lista de precios del menú.

MÁS INFORMACIÓN



Quickselect.