C-4.13 Suppose we are given two sequences A and B of n elements, possibly containing duplicates, on which a total order relation is defined. Describe an efficient algorithm for determining if A and B contain the same set of elements (possibly in different orders). What is the running time of this method?

|  |  |
| --- | --- |
| Agorithm isSameElements(A, B)  if A.size() <> B.size() then  return false    \_inPlaceQuickSort(A, 0, A.size())  \_inPlaceQuickSort(B, 0, B.size())    for i<-0 to A.size() do  p <- A.elementAtRank(i)  q <- B.elementAtRank(i)  if p.element() <> q.element() then  return false    return true    \_inPlaceQuickSort(S, l, h)  if l < h then  k <- \_inPlacePartition(S, l, h)  \_inPlaceQuickSort(S, l, k-1)  \_inPlaceQuickSort(S, k+1, h)  \_inPlacePartition(S, l, h)  p <- randomInt(l, h)    S.swapElement(S.atRank(l), S.atRank(p))    j <- l+1  k <- h  while j < k do  while k > j /\ S.elementAtRank(k) >= S.elementAtRank(l) do  k <- k - 1    while j < k /\ S.elementAtRank(j) <= S.elementAtRank(l) do  j <- j + 1    if j < k then  S.swapElement(S.elementAtRank(j), S.elementAtRank(k))    S.swapElement(S.elementAtRank(k), S.elementAtRank(l)    return k | O(1)  O(1)  O(nlogn)  O(n)  O(n)  O(n)  O(n)  O(n)  O(1)  Total running time is: O(nlogn) |

R-5.4 Characterize each of the following recurrence equations using the master method (assuming that T(n) = c for n < d, for constant c > 0 and d  1).

a. T(n) = 2T(n/2) + logn

b. T(n) = 8T(n/2) + n2

c. T(n) = 16T(n/2) + (nlogn)4

d. T(n) = 7T(n/3) + n

e. T(n) = 9T(n/3) + (n3logn)

A picture containing white

Description automatically generated

A close up of text on a white background

Description automatically generated