

Algorithm represent (A, B)

~~B := new Priority Queue~~

~~B := new P~~

~~A := removeDup(A)~~

~~B := removeDup(B)~~

S := new Sequence.

while $\neg A.empty \vee \neg B.empty$ do

if $A.first() = B.first()$ then

$B.remove(B.first())$

else if $A.first() < B.first()$ then

if $S.last() = A.first()$

$S.insert(A.first())$

$A.remove(A.first())$

else if $A.first() > B.first()$ then

if $S.last() = B.first()$

$S.insert(B.first())$

$B.remove(B.first())$

R-4.9

if the middle element of sorted 's' is selected \rightarrow as pivot
size of both sides L & R will be $\leq n/4 \rightarrow$ height $\rightarrow \log_{4/3} n$
& running time $O(n)$

$\hookrightarrow O(\log_{4/3} n) \rightarrow O(n \log n)$

C-4.10

Algorithm electionVotes(s)

D: s new Dictionary with Key integer & value Integer.

For i: 0 to s.size() do

if d.containsKey(s[i])

D[s[i]]++

else

d.add(s[i], 1)

max := 0

winnerId := integer

For each player in d do

if player.votes > max

max ← player.votes

winnerId ← player.ID

return winnerId