

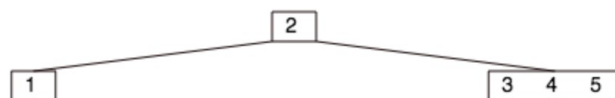
## Assignment 8

R-3.8 Is the multi way search tree of Figure 3.17a a (2,4) tree? Justify your answer.

No, because the external nodes are in different depth

R-3.10 A certain Professor Amongus claims that a (2,4) tree storing a set of items will always have the same structure, regardless of the order in which the items are inserted. Show that Professor Amongus is wrong.

Inputs: 1 2 3 4 5



Inputs: 5 4 3 2 1



C-4.11 Suppose we are given an  $n$ -element sequence  $S$  such that each element in  $S$  represents a different vote in an election, where each vote is given as an integer representing the ID of the chosen candidate. Suppose we know who the candidates are and the number of candidates running is  $k < n$ . Describe an  $O(n \log k)$ -time algorithm for determining who wins the election.

Algorithm `getElectionWinner(S)`

Input: Vote sequence  $S$

Output: Winner candidate Id

dictionary  $\leftarrow$  Dictionary with AVL tree implementation

iterator  $\leftarrow S.elements()$

while iterator.hasNext() do

    candidateId  $\leftarrow$  iterator.nextObject()

    count  $\leftarrow$  dictionary.removeElement(candidateId)

    if count = NO\_SUCH\_KEY then

        dictionary.insertItem(candidateId, 1)

    else

        dictionary.insertItem(candidateId, count + 1)

maxCount  $\leftarrow$  0

winnerCandidateId  $\leftarrow$  NULL

iterator  $\leftarrow$  dictionary.keys()

while iterator.hasNext() do

    candidateId  $\leftarrow$  iterator.nextObject()

    count  $\leftarrow$  dictionary.findElement(candidateId)

    if maxCount < count then

        maxCount  $\leftarrow$  count

        winnerCandidateId  $\leftarrow$  candidateId

return winnerCandidateId

C-4-22 Let A and B be two sequences of n integers each. Given an integer x, describe an  $O(n \log n)$ -time algorithm for determining if there is an integer a in A and an integer b in B such that  $x = a + b$ .

Algorithm checksum(A, B, x)

Input: Sequence A and B, and sum x

Output: Whether A and B contain integers whose sum is equal to x

Dictionary  $\leftarrow$  Dictionary with AVL tree implementation

iterator  $\leftarrow$  A.elements()

while iterator.hasNext() do

    a  $\leftarrow$  iterator.nextObject()

    dictionary.insertItem(a, a)

iterator  $\leftarrow$  B.elements()

while iterator.hasNext() do

    b  $\leftarrow$  iterator.nextObject()

    key  $\leftarrow$  x - b

    a  $\leftarrow$  dictionary.findElement(key)

    if a  $\neq$  NO\_SUCH\_KEY

        return true

return false