Chanh Dao Le – 986178

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 <- Dictionary with AVL tree implementation
        iterator <- S.elements()
        while iterator.hasNext() do
                candidateId <- iterator.nextObject()
                count <- dictionary.removeElement(candidateId)
                if count = NO_SUCH_KEY then
                        dictionary.insertItem(candidateId, 1)
                else
                        dictionary.insertItem(candidateId, count + 1)
        maxCount <- 0
        winnerCandidateId <- NULL
        iterator <- dictionary.keys()
        while iterator.hasNext() do
                candidateId <- iterator.nextObject()
                count <- dictionary.findElement(candidateId)
                if maxCount < count then
                        maxCount <- count
                        winnerCandidateId <- candidateId
        return winnderCandidateId
```

This study source was downloaded by 100000849104862 from CourseHero.com on 06-29-2022 14:16:16 GMT -05:00

https://www.coursehero.com/file/32043297/Ass8docx/

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 <- Dictionary with AVL tree implementation
        iterator <- A.elements()
        while iterator.hasNext() do
                a <- iterator.nextObject()
                dictionary. insertItem(a, a)
        iterator <- B.elements()
        while iterator.hasNext() do
                b <- iterator.nextObject()
                key <- x – b
                a <- dictionary.findElement(key)
                if a ¬= NO_SUCH_KEY
                        return true
        return false
```

This study source was downloaded by 100000849104862 from CourseHero.com on 06-29-2022 14:16:16 GMT -05:00

https://www.coursehero.com/file/32043297/Ass8docx/