Name: Pheakdey Luk
ID: 986591

Assignment 8

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

Answer:

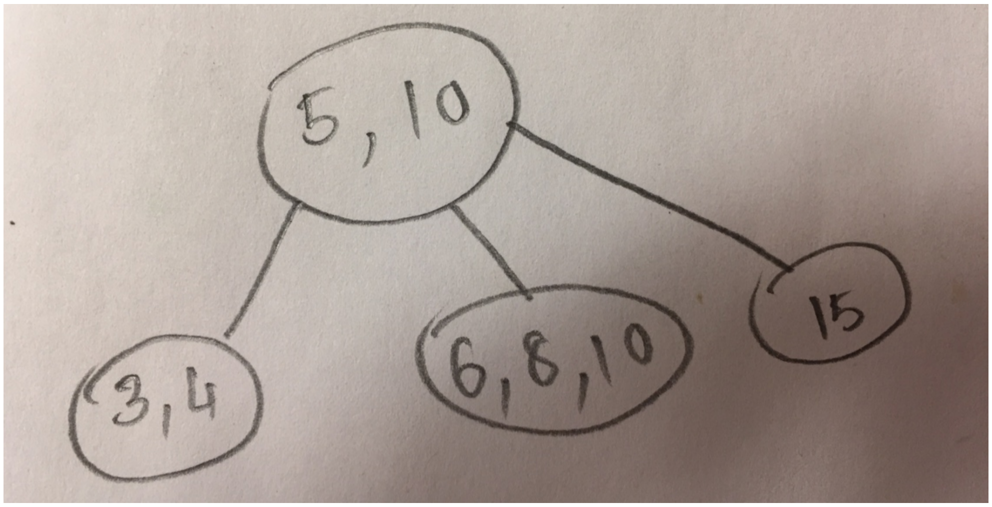
No, it is not a (2, 4) tree because (2,4) tree has following properties

1. Node-Size Property: every internal node has at most four children
2. Depth Property: all the external nodes have the same depth

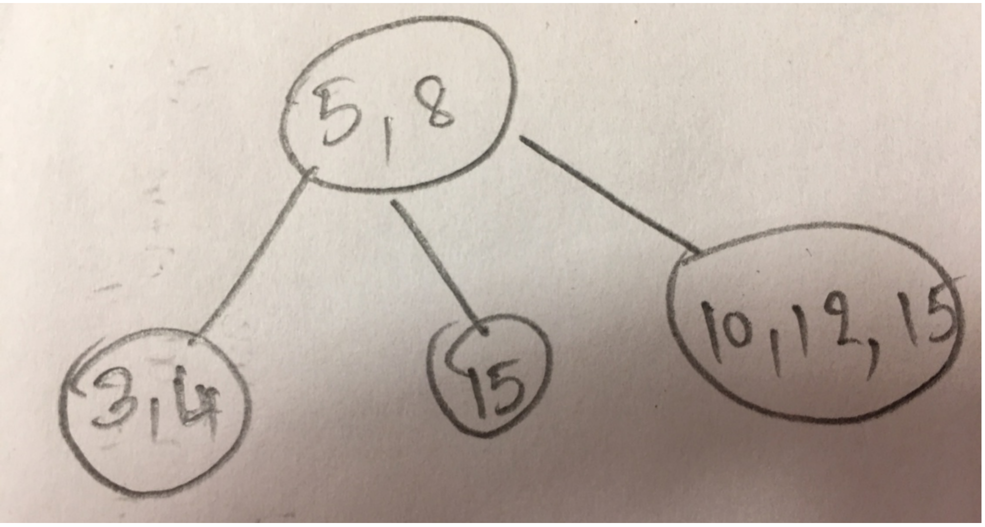
So, the multi way search tree of Figure 3.17a is not a (2,4) tree because
As the external nodes don't have the same depth of the multi way search tree of Figure 3.17a, It can be a (2, 4) tree.

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.

if we insert following items in this order we get below structure { 4, 6, 12, 15, 3, 5, 10, 8 }



Now, if we change the order(reverse order): { 8, 10, 5, 3, 15, 12, 6, 4 },we get a different structure for same items



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.

Answer:

```
Algorithm electionWinner(S, C)
    Input: Sequence S of votes and Sequence SC of candidates
    Ouput:Id of winner candidate

    D<-new Dictionary(Hastable)
    for each id of C do
        D.insertItem(id,0)
    for each y of S do
        cnt<-D.findElement(x)
        D.insertItem(y,cnt + 1)
    winnerId <- -1
    for each (id,cnt) of D.items() do
        if cnt > winnerId then
            winnerId<-id
            winnerCount<-cnt

    return winnerId
```

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.

Answer:

```
Algorithm isSumEqual(A,B,x)
    Input: Two Sequence A, B and integer x
    Ouput: true or false if such two intergers of A and B makes x
    T <- insert A into Redblack tree
    for each y of B do
        if T.findElement(x-y) != NULL then
            return true

    return false
```

8. A. Suppose the only available sorting algorithm in the library for an array-based Sequence is QuickSort(S, C); this would be similar to the Java class library for sorting an ArrayList. However, Quick Sort is not a stable sort but you need a sort that is stable. You are to write a pseudo code function **StableSort(S)** for a Sequence S of Objects (containing duplicates) using QuickSort and the built-in comparator <.

Hint: Change the key (add an element to the key, i.e., make it a pair) and create a new comparator which would include <.