



University of  
South Australia

UniSA STEM

## INFS 2042 Data Structures Advanced

### Practical 6

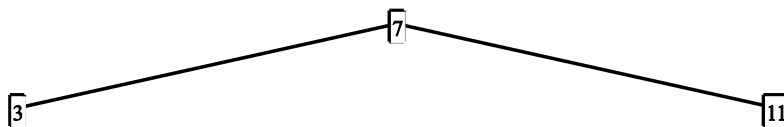
Self-Balancing Trees – 2-3 Trees, B-Trees, B+ Tree, 2-3-4 Trees

**Warning:** This material has been reproduced and communicated to you by or on behalf of the University of South Australia in accordance with section 113P of the Copyright Act 1968 (Act). The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Complete the tasks below. Only the task labelled as “Continuous Assessment Task” will be assessed.

#### Task 1 (Not Assessed)

Show how the following 2-3 tree would change after inserting each of the following values one at a time: 2, 6, 10.



#### Task 2 (Not Assessed)

Show the 2-3 tree that would store the words in the sentence “Now is the time for all good people to come to the aid of others”. Presume the capital letters precedes lower case letters. What would be the 2<sup>nd</sup> virtual node (a temporary node that stores 3 items before splitting)? The third virtual node? The fourth virtual node?

#### Task 3 (Not Assessed)

Draw a B-tree with order 5 that stores the sequence of integers:

20, 30, 8, 10, 15, 18, 44, 26, 28, 23, 25, 43, 55, 36, 44, 39.

#### Task 4 (Not Assessed)

Draw the B+ tree that would be formed by inserting the integers shown in Task 3.

#### Task 5 (Not Assessed)

Construct and draw a 2-3-4 Tree by inserting the following items into it:

13, 24, 34, 23, 98, 45, 75, 46, 30, 12, 2

Note that in a 2-3-4 Tree the split happens whenever a 4-node is encountered while traversing the tree (even if it is not the node where the item should be inserted).

## Task 6 (Not Assessed)

Build a 2-3-4 tree to store the words in the sentence “Now is the time for all good people to come to the aid of others”.

## Continuous Assessment Task (3 marks)

You are provided with an incomplete Java code for the implementation of a 2-3-4 Tree.

For this task you are asked to complete the implementation of the following three functions:

- Node.java
  - `Node<T> getNextNode(T value)`
    - This function should return the child `Node` to traverse next based on the given value.
    - Check how this function is used in the `find()` function of the `TwoThreeFourTree` class.
    - The key parts of this function:
      - Iterate through the `itemArray` until the item is less than the target value (you can use `compareTo` for this)
        - Get the correct link in the `childArray` and return it.
      - If the end of the `itemArray` is reached, return the last link in the `childArray`.
      - Hint: You may presume the item is not present in this node.
  - `int insertItem(T item)`
    - This function should insert the `item` at the correct location within this node and return the index of the inserted item.
    - Check how this function is used in the `split()` function of the `TwoThreeFourTree` class.
    - The key parts of this function:
      - Iterate through the `itemArray` and inserting at the target location.
      - Update the number of items in the node.
      - Return the index where the item was inserted.
      - Return -1 if the item already exists in this node.
      - Hint: You may presume this is a leaf node, and it is not full.
- TwoThreeFourTree.java
  - `boolean insert(T item)`
    - This function should insert the `item` into the 2-3-4 tree.
    - It should handle traversing the tree and splitting any 4-Nodes that are found along the way.
    - It should return `true` if the item was successfully inserted or `false` if the item was already found in the tree.
    - The key parts of this function:
      - Traverse the tree nodes starting from the root
        - If a node is full, split before continuing.
        - If a leaf node is reached, stop traversing and insert the item.

### Sample Output:

```
Values: {13, 24, 45}, level: {0} , IsLeaf {false}
Values: {2, 12}, level: {1} , IsLeaf {true}
Values: {23}, level: {1} , IsLeaf {true}
Values: {30, 34}, level: {1} , IsLeaf {true}
Values: {46, 75, 98}, level: {1} , IsLeaf {true}

Values: {Happy}, level: {0} , IsLeaf {false}
Values: {Charla, Erza}, level: {1} , IsLeaf {false}
Values: {Luxus, Natsu, Sanji}, level: {1} , IsLeaf {false}
Values: {A black cat whoes name I forgot, Brook}, level: {2} , IsLeaf {true}
Values: {Chopper, Elfman}, level: {2} , IsLeaf {true}
Values: {Franky, Gageel, Gray}, level: {2} , IsLeaf {true}
Values: {Jimbe, Lucy, Luffy}, level: {2} , IsLeaf {true}
Values: {Nami}, level: {2} , IsLeaf {true}
Values: {Robin}, level: {2} , IsLeaf {true}
Values: {Zoro}, level: {2} , IsLeaf {true}
```