



# B-Trees

---

*NDBI007: Assignment 4*

# Task 1: Non-Redundant B-Tree

---

- ❖ Suppose a *non-redundant B-tree* of degree  $m \in \langle 3,5 \rangle$ 
  - ❖ Determine the *minimum number of children* and the *minimum number of keys* at each node except the root of the tree
  - ❖ *Insert only the first 10 assigned values* into the tree in the given order
  - ❖ Finally, *delete the first 4 inserted* values from the tree (in an arbitrary order)
- ❖ *Illustrate* the state of the tree *each time a node is split or merged* and also the *final result*
  - ❖ I.e., if you are only trivially inserting or deleting a key, just describe what happened
- ❖ Or, you may *implement* the B-tree *and log* all events (i.e., submit the source code and, e.g., makefile)
  - ❖ The permitted languages are Java, Python, C, C++, and Swift
- ❖ **Points: 1**

# Task 2: B<sup>+</sup>-Tree

---

- ❖ Suppose a *B<sup>+</sup>-tree* of degree  $m \in \langle 3,6 \rangle$ 
  - ❖ Determine the *minimum number of children* and the *minimum number of keys* at each node except the root of the tree
  - ❖ *Insert all 15 assigned values* into the tree in the given order
  - ❖ Finally, *delete the first 4 inserted* values from the tree (in an arbitrary order)
- ❖ *Illustrate* the state of the tree *each time a node is split or merged* and also the *final result*
  - ❖ I.e., if you are only trivially inserting or deleting a key, just describe what happened
- ❖ Or, you may *implement* the B<sup>+</sup>-tree *and log* all events (i.e., submit the source code and, e.g., makefile)
  - ❖ The permitted languages are Java, Python, C, C++, and Swift
- ❖ **Points: 1**

# Task 3: B\*-Tree

---

- ❖ Suppose a *B\*-tree* of degree  $m \in \langle 4,6 \rangle$ 
  - ❖ Determine the *minimum number of children* and the *minimum number of keys* at each node except the root of the tree
  - ❖ *Insert all 15 assigned values* into the tree in the given order
  - ❖ Finally, *delete the first 4 inserted* values from the tree (in an arbitrary order)
- ❖ *Illustrate* the state of the tree *each time a node is split or merged* and also the *final result*
  - ❖ I.e., if you are only trivially inserting or deleting a key, just describe what happened
- ❖ Or, you may *implement* the B\*-tree *and log* all events (i.e., submit the source code and, e.g., makefile)
  - ❖ The permitted languages are Java, Python, C, C++, and Swift
- ❖ **Points: 1**

# Bonus Task 4: Redundant B-Tree

---

- ❖ Solve the Task 1 for redundant B-tree
- ❖ In addition, *illustrate an example* in which case you will use a *redundant B-tree* and another example illustrating a suitable utilization of *non-redundant B-tree*
- ❖ **Points: 1**