# Data Structures 1

NTIN066

Jirka Fink

Department of Theoretical Computer Science and Mathematical Logic
Faculty of Mathematics and Physics
Charles University in Prague

Summer semester 2025/26

Last change on March 10, 2026

# (a,b)-Tree (Bayer, McCreight, 1972)

## Definition

An $(a, b)$-tree is a search tree that satisfies the following properties:

1. The parameters $a$ and $b$ are integers such that $a \geq 2$ and $b \geq 2a - 1$.
2. All internal nodes, except for the root, must possess at least $a$ and at most $b$ children.
3. The root may have at most $b$ children.
4. All leaves are required to be at the same depth.

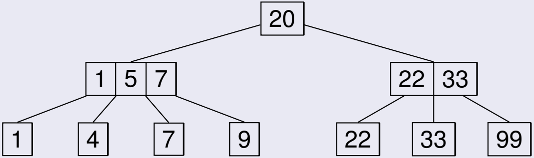
# (a,b)-Tree (Bayer, McCreight, 1972)

## Definition

An $(a, b)$-tree is a search tree that satisfies the following properties:

1. The parameters $a$ and $b$ are integers such that $a \geq 2$ and $b \geq 2a - 1$.
2. All internal nodes, except for the root, must possess at least $a$ and at most $b$ children.
3. The root may have at most $b$ children.
4. All leaves are required to be at the same depth.

## Example: (2,4)-Tree



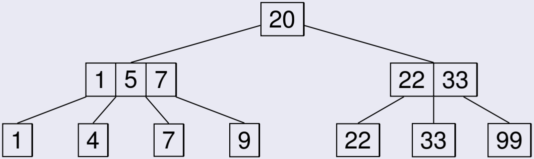
# (a,b)-Tree (Bayer, McCreight, 1972)

## Definition

An $(a, b)$-tree is a search tree that satisfies the following properties:

1. The parameters $a$ and $b$ are integers such that $a \geq 2$ and $b \geq 2a - 1$.
2. All internal nodes, except for the root, must possess at least $a$ and at most $b$ children.
3. The root may have at most $b$ children.
4. All leaves are required to be at the same depth.

## Example: (2,4)-Tree




## Operations

- Describe operations FIND, INSERT and DELETE and determine their complexity.
- Why the condition $b \geq 2a - 1$ is necessary?

## Modified nodes

Estimate the number of modified nodes during a sequence of $n$ operations INSERT into an empty tree.

# Exercises

## Modified nodes

Estimate the number of modified nodes during a sequence of $n$ operations INSERT into an empty tree.

## Build

Given $n$ elements, how fast can be build an (a,b)-tree?

# Exercises

## Modified nodes

Estimate the number of modified nodes during a sequence of $n$ operations INSERT into an empty tree.

## Build

Given $n$ elements, how fast can be build an (a,b)-tree?

## Using a Smaller Amount of Memory

- The number of children of a node is between $a$ and $b$, where $b \geq 2a - 1$.
- If a node has only $a$ children, approximately half of its memory may remain unused.
- Is it possible to modify the Insert and Delete operations so that at most $1/3$ of a node's memory is unused?
- Specifically, describe the operations in a variant of the $(a, b)$-tree where $a \approx \frac{2}{3}b$.

# Modification of (a,b)-trees

## Using a Smaller Amount of Memory

- The number of children of a node is between $a$ and $b$, where $b \geq 2a - 1$.
- If a node has only $a$ children, approximately half of its memory may remain unused.
- Is it possible to modify the Insert and Delete operations so that at most $1/3$ of a node's memory is unused?
- Specifically, describe the operations in a variant of the $(a, b)$-tree where $a \approx \frac{2}{3}b$.

**Question**

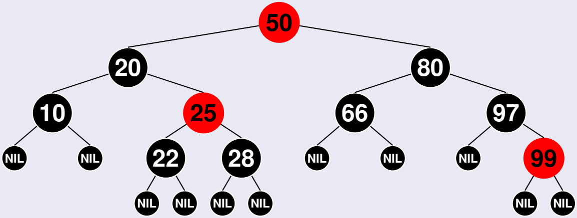
Can we parallelize operations FIND, INSERT and DELETE in (a,b)-trees?

# Red–Black Trees (Guibas, Sedgewick, 1978)

## Definition

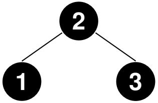
1. A binary search tree with elements stored in all nodes
2. Each node is either black or red
3. All paths from the root to the leaves contain the same number of black nodes
4. The parent of a red node must be black
5. Leaves are black

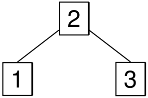
## Example



Each black node corresponds to one node of a (2,4)-tree

- Node without red children

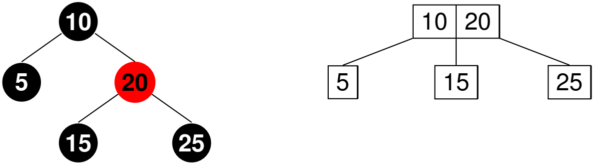


Each black node corresponds to one node of a (2,4)-tree

- Node without red children

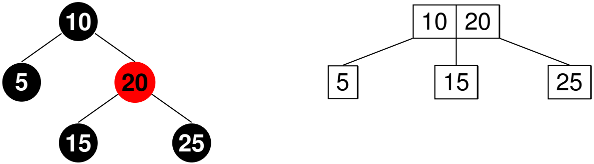

- Node with one red child



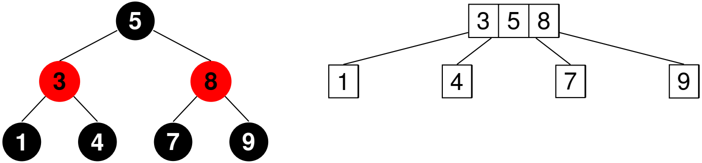
Each black node corresponds to one node of a (2,4)-tree

- Node without red children


- Node with one red child




- Node with two red children



# (a,b)-Tree: Number of modified nodes

## Theorem (Huddleston, Mehlhorn, 1982)

The amortized number of splits and merges in any sequence of $k$ operations INSERT and DELETE in an $(a, 2a)$-tree is $\mathcal{O}(1)$, and the total number is $\mathcal{O}(n + k)$.

# (a,b)-Tree: Number of modified nodes

## Theorem (Huddleston, Mehlhorn, 1982)

The amortized number of splits and merges in any sequence of $k$ operations INSERT and DELETE in an $(a, 2a)$-tree is $\mathcal{O}(1)$, and the total number is $\mathcal{O}(n + k)$.

## Proof

- The potential of a node $u$ depends on the number of its children as follows:

| Children | $a-1$ | $a$ | $a+1$ | $\dots$ | $2a-1$ | $2a$ | $2a+1$ |
|---|---|---|---|---|---|---|---|
| $\Phi(u)$ | 2 | 1 | 0 | $\dots$ | 0 | 2 | 4 |

- The potential of the tree is the sum of the potentials of its nodes.

# (a,b)-Tree: Number of modified nodes

## Theorem (Huddleston, Mehlhorn, 1982)

The amortized number of splits and merges in any sequence of $k$ operations INSERT and DELETE in an $(a, 2a)$-tree is $\mathcal{O}(1)$, and the total number is $\mathcal{O}(n+k)$.

## Proof

- The potential of a node $u$ depends on the number of its children as follows:

| Children | $a-1$ | $a$ | $a+1$ | $\dots$ | $2a-1$ | $2a$ | $2a+1$ |
|---|---|---|---|---|---|---|---|
| $\Phi(u)$ | 2 | 1 | 0 | $\dots$ | 0 | 2 | 4 |

- The potential of the tree is the sum of the potentials of its nodes.
- Changes in potential when splitting node $u$ with parent $p$ are as follows:
  - If $u$ has a potential of 4, it is split into two nodes with potentials 0 and 1.
  - The potential of node $p$ increases by at most 2.
  - The potential decreases by at least 1, which covers the cost of the split.

# (a,b)-Tree: Number of modified nodes

## Theorem (Huddleston, Mehlhorn, 1982)

The amortized number of splits and merges in any sequence of $k$ operations INSERT and DELETE in an $(a, 2a)$-tree is $\mathcal{O}(1)$, and the total number is $\mathcal{O}(n + k)$.

## Proof

- The potential of a node $u$ depends on the number of its children as follows:

| Children | $a-1$ | $a$ | $a+1$ | $\dots$ | $2a-1$ | $2a$ | $2a+1$ |
|---|---|---|---|---|---|---|---|
| $\Phi(u)$ | 2 | 1 | 0 | $\dots$ | 0 | 2 | 4 |

- The potential of the tree is the sum of the potentials of its nodes.
- Changes in potential when splitting node $u$ with parent $p$ are as follows:
  - If $u$ has a potential of 4, it is split into two nodes with potentials 0 and 1.
  - The potential of node $p$ increases by at most 2.
  - The potential decreases by at least 1, which covers the cost of the split.
- Changes in potential when merging nodes $u$ and $u'$ with parent $p$ are as follows:
  - If $\Phi(u) = 2$ and $\Phi(u') = 1$, the merged node has a potential of 0.
  - The potential of node $p$ increases by at most 1.
  - The potential decreases by at least 2, which covers the cost of the merge.

# (a,b)-Tree: Number of modified nodes

## Theorem (Huddleston, Mehlhorn, 1982)

The amortized number of splits and merges in any sequence of $k$ operations INSERT and DELETE in an $(a, 2a)$-tree is $\mathcal{O}(1)$, and the total number is $\mathcal{O}(n+k)$.

## Proof

- The potential of a node $u$ depends on the number of its children as follows:

| Children | $a-1$ | $a$ | $a+1$ | $\dots$ | $2a-1$ | $2a$ | $2a+1$ |
|---|---|---|---|---|---|---|---|
| $\Phi(u)$ | 2 | 1 | 0 | $\dots$ | 0 | 2 | 4 |

- The potential of the tree is the sum of the potentials of its nodes.
- Changes in potential when splitting node $u$ with parent $p$ are as follows:
  - If $u$ has a potential of 4, it is split into two nodes with potentials 0 and 1.
  - The potential of node $p$ increases by at most 2.
  - The potential decreases by at least 1, which covers the cost of the split.
- Changes in potential when merging nodes $u$ and $u'$ with parent $p$ are as follows:
  - If $\Phi(u) = 2$ and $\Phi(u') = 1$, the merged node has a potential of 0.
  - The potential of node $p$ increases by at most 1.
  - The potential decreases by at least 2, which covers the cost of the merge.
- Adding, deleting, and moving a node can increase the potential by at most 2, but these operations are performed at most once.
- Since $0 \leq \Phi \leq 4n$, the total decrease in potential across all operations is $\mathcal{O}(n)$.