

# Červeno-čierny strom

## 5. projekt, ITY

Katarína Mečiarová

May 4, 2024

- **Algoritmy sú jedným zo základných prvkov spracovania informácií**

- **Čo je to červeno-čierny strom?**

Od vzniku, cez praktické využitie až do končín teórie

# "...the rest is history"

- **Kto? Rudolf Bayer**  
symetrický binární B-strom  
Leo J. Guibase, Robert Sedgewick
- **Kedy? 1972,**  
premenované 1978, ... 1993, 1999
- **Ako? To si rozoberieme na drobné v nasledujúcom slajde**



# "...the rest is history"

**1972**

R. Bayer prišiel s novou dátovou štruktúrou, tvoriacou "dokonale vyrovnané stromy"

**1978**

Guibas a Sedgewick spomenuli štruktúru pod menom "červeno-čierny strom"  
význam výberu červenej farby bol čisto pragmatický

**1993**

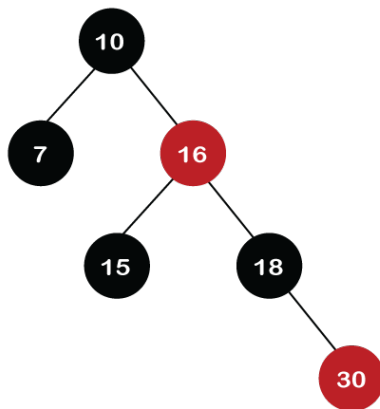
**Arne Andersson** predstavil svetu "LL strom" - zjednodušenie a vyrovnanie

**1999**

**Chris Okasaki** objavil nový spôsob a tak pracoval len so 4 nevyrovnanými prípadmi

...  
Čo s tým?

- funkcionálne programovanie
- real-time procesy (RTC)
- geometrické algoritmy
  
- asociatívne reťazce
- množiny
- slovníky



# Definícia štruktúry

- 1 Každý vrchol je buď čierny alebo červený.
- 2 Koreň, materský vrchol je čierny.
- 3 Ak je vrchol červený, obe jeho deti sú čierne.  
Žiadny červený vrchol nemá žiadne červené dieťa.
- 4 Každá cesta z (akéhokoľvek) vrcholu do (akéhokoľvek) listu má rovnaký počet čiernych vrcholov.

Teda ak vrchol **N** má presne **1 dieťa**, tak to dieťa **musí byť červené**.

Ak by bolo dieťa čierne, jeho NIL potomkovia by sa nachádzali v inej (čiernej) hĺbke ako NIL potomok N, čím by sa porušila požiadavka 4.

Zapisujeme pomocou Landauovej notácie.

- pridať vrchol/list
- odstrániť
- preoznačiť (zmena farby vrcholu)
- zlúčenie stromov
- rozdelenie stromu
- rotácie
  - pravá
  - ľavá

# Pseudokód pridatia prvku

```
INSERT(T, n)
  y = T.NIL
  temp = T.root

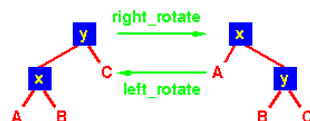
  while temp != T.NIL
    y = temp
    if n.data < temp.data
      temp = temp.left
    else
      temp = temp.right
  n.parent = y
  if y==T.NIL
    T.root = n
  else if n.data < y.data
    y.left = n
  else
    y.right = n

  n.left = T.NIL
  n.right = T.NIL
  n.color = RED
  INSERT_FIXUP(T, n)
```



# Pseudokód rotácie





```
left_rotate( Tree T, node x ) {  
    node y;  
    y = x->right;  
    /* Turn y's left sub-tree into x's right sub-tree */  
    x->right = y->left;  
    if ( y->left != NULL )  
        y->left->parent = x;  
    /* y's new parent was x's parent */  
    y->parent = x->parent;  
    /* Set the parent to point to y instead of x */  
    /* First see whether we're at the root */  
    if ( x->parent == NULL ) T->root = y;  
    else  
        if ( x == (x->parent)->left )  
            /* x was on the left of its parent */  
            x->parent->left = y;  
        else  
            /* x must have been on the right */  
            x->parent->right = y;  
    /* Finally, put x on y's left */  
    y->left = x;  
    x->parent = y;  
}
```



- AVL strom
- AA strom
- B strom, B+ strom, ...
- T strom
- WAVL strom
- Splay strom
- Scapegoat strom

Ďakujem za Vašu pozornosť



-  Bayer, R.: *Symmetric binary B-Trees: Data structure and maintenance algorithms*. Acta Informatica, 1972, doi: 10.1007/BF00289509.
-  Cormen, T. H.: *Introduction to Algorithms*. Acta Informatica, 2001, ISBN 978-0-262-03293-3.
-  Focko, M.: Použití červeno-černých stromů. *IB002: Algorithms*, 11.12.2022.  
URL <https://www.fi.muni.cz/~xfocko/kb/ib002/rb-trees/applications>
-  Morris, J.: 8.2 Red-Black Trees. *Data Structures and Algorithms*, 1998.  
URL [https://www.eecs.umich.edu/courses/eecs380/ALG/red\\_black.html](https://www.eecs.umich.edu/courses/eecs380/ALG/red_black.html)