



# Semestrálna práca 2

Algoritmy a údajové štruktúry 2

Lukáš Urbaník

5ZII11

## Popis riešenia

Na implementáciu registra vozidiel a vodičských preukazov sme použili údajové štruktúry BTree (3x), pre vodičské preukazy, evidenčné čísla vozidiel a VIN čísla vozidiel, a HeapFile pre vozidlá. Operácie mazania nie sú implementované.

### Počet prístupov do súboru

N- výška stromu

#### Vodičské preukazy:

**Vyhľadaj:** najlepší prípad – 1 prístup, záznam je v koreni.

Najhorší prípad – záznam je v liste N prístupov.

**Vlož:** najlepší prípad – list má voľné miesto. Načítanie blokov od koreňa po list N a zápis bloku + 1.

Najhorší prípad – všetky bloky sú plné. Rozdelenie blkov na dva + ich zápis + čítanie otca, zápis nového koreňa.  $3 \cdot N + 1$ .

**Uprav:** najlepší prípad – prvok je v liste, načítame jeden blok a zapíšeme blok s upraveným záznamom, 2 prístupy.

Najhorší prípad – prvok je v liste, teda  $N + 1$  (zápis upraveného bloku) prístupov.

#### Vozidlá:

**Vyhľadaj podľa VIN:** najlepší prípad – 1 prístup, záznam je v koreni.

Najhorší prípad – záznam je v liste N prístupov.

**Vyhľadaj podľa EČ:** najlepší prípad – 1 prístup, záznam je v koreni.

Najhorší prípad – záznam je v liste N prístupov.

**Vlož:** najlepší prípad – list má voľné miesto. Načítanie blokov od koreňa po list N a zápis bloku + 1 pre strom s ECV,  $N + 1$  pre strom s VIN a 1 pre HeapFile.

Najhorší prípad – všetky bloky sú plné. Rozdelenie blkov na dva + ich zápis, zápis nového koreňa.  $3 \cdot N + 1$  pre strom s ECV,  $3 \cdot N + 1$  pre strom s VIN a 1 pre HeapFile.

**Uprav:** najlepší prípad – prvok je v liste, načítame jeden blok a zapíšeme blok s upraveným záznamom, 2 prístupy.

Najhorší prípad – prvok je v liste, teda  $N + 1$  (zápis upraveného bloku) prístupov.