

# SBD - projekt 2

Paweł Cichowski

## Wprowadzenie

Celem projektu było zaimplementowanie pliku o organizacji indeksowej, korzystając ze struktury B-drzewa.

W programie zaimplementowane zostały operacje: dodawania rekordu, usuwania, aktualizacji.

Operacje na pliku indeksowym są przeprowadzane z wykorzystaniem stron - każda strona zawiera indeksy opisujące dane oraz adresy stron potomnych. Strony są buforowane w pamięci w wektorze, który może zawierać do  $h$  stron, czyli jedną gałąź drzewa.

Operacje na pliku danych zasymulowane są jako blokowe - odczyt lub zapis do dysku są wykonywane tylko przy sytuacji "cache miss". w pamięci buforowany jest jeden blok rekordów.

## Opis działania programu

### Parametry pozycyjne

Program obsługuje kilka parametrów pozycyjnych, pozwalających wybrać tryb pracy:

- -a, --automatic <path> - program przyjmuje instrukcje z pliku testowego, opisanego w kolejnym punkcie
- -i, --interactive - program przyjmuje instrukcje z konsoli, w takim samym formacie co plik testowy
- -v, --verbose - program opisuje każdą wykonaną instrukcję

Program generuje również plik z logami w formacie csv. Zawiera on kolejne wykonane instrukcje i czas ich wykonania w milisekundach.

## Plik testowy

Format pliku testowego:

```
rząd drzewa
maksymalny indeks losowo generowanych rekordów
instrukcja 1
instrukcja 2
...
```

Dostępne instrukcje:

- i - insert rekordu o losowym kluczu
- i <key> - insert rekordu o podanym kluczu
- ip <key> - insert, po operacji wyprintowanie drzewa
- d <key> - delete rekordu o podanym kluczu
- dp <key> - delete, po operacji wyprintowanie drzewa
- u <key> <identity> <name> <surname> <age> - update danych rekordu o podanym kluczu
- p - printuje całe drzewo
- end - koniec działania programu

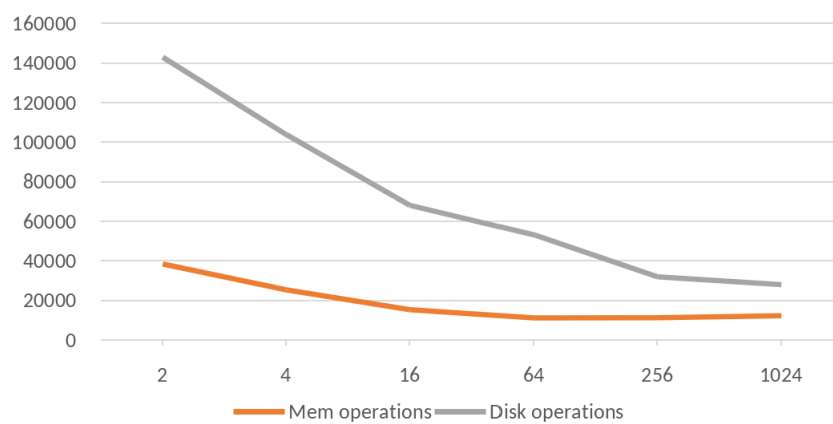
## Wyniki eksperymentów

Eksperyment polegał na dodaniu do pliku  $n$  rekordów i zmierzeniu liczby operacji dyskowych, pamięciowych oraz czasu działania dla różnych wartości rzędu B-drzewa.

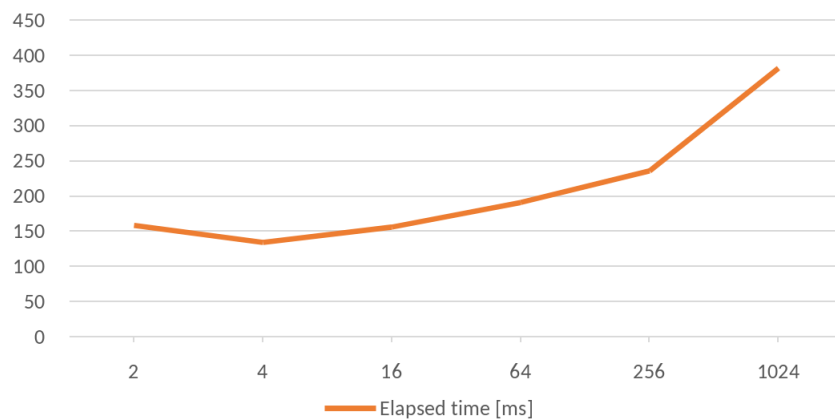
Danymi wejściowymi testów były: liczba rekordów: 10.000 oraz 1.000.000 dla rzędów drzewa: 2, 4, 16, 64, 256, 1024.

### 10.000 rekordów

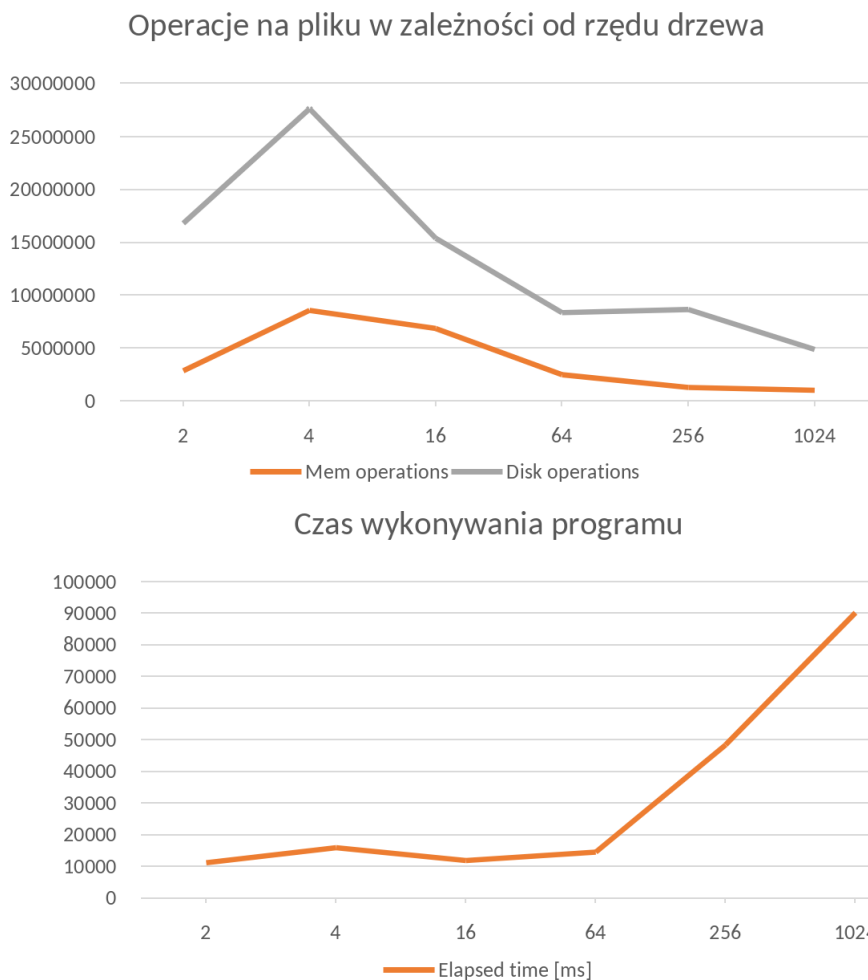
Operacje na pliku w zależności od rzędu drzewa



Czas wykonywania programu



## 1.000.000 rekordów



## Wnioski

Na podstawie przedstawionych wykresów postawione zostały poniższe wnioski:

- Większy rząd drzewa oznacza mniej zapisów do dysku, natomiast prowadzi do zwiększenia złożoności czasowej - większe strony są dłużej zapisywane.
- Większy rząd oznacza również korzystniejszy stosunek między operacjami w pamięci a dyskowymi. Większe strony to częstszy "cache hit"

- Dla dużej liczby rekordów widoczna jest optymalna wartość rzędu drzewa - 64. Zachowuje kompromis między czasem działania a liczbą operacji dyskowych.