# Sprawozdanie z projektu drugiego - Struktury Baz Danych

Piotr Sieński

31 grudnia 2022

## 1 Opis projektu

### 1.1 Zastosowana metoda

Do realizacji zadania wykorzystane zostało B-drzewo. Buforowanie pamieci jest zrealizowane poprzez mechanizm cache z polityka Least Recently Used - pamieć cache ma stały rozmiar i w momencie przepełnienia do dysku przepisywana jest strona która była najdawniej uzywana. Cache zrealizowane jest jako odrebna warstwa logiki programu i działa poprzez trzymanie listy stron w pamieci. Gdy nastepuje dostep do strony jest ona przesuwana na poczatek listy. Oprócz operacji dodawania, usuwania i aktualizacji rekordu zaimplementowano również operacje reorganizacji plików - zarówno pliku indeksowego jak i pliku z danymi. Kiedy zachodzi reorganizacja rekordy z końca pliku sa przepisywane w puste miejsca (oznaczone jako usuniete) i rozmiar pliku jest zmniejszany. Dodaktowo po reorganizacji pliku z danymi nastepuje aktualizacja wskaźników na dane w pliku indeksowym.

### 1.2 Specyfikacja formatu pliku testowego

Typem rekordów sa walce o danym promieniu i wysokości. Plik testowy jest plikiem tekstowym zawierajacym ciag instrukcji dodawania, usuwania i aktualziacji. Każda instrukcja znajduje sie w nowej linii i ma format :

```
instrukcja argumenty
```

możliwe instrukcje to

```
i - wstawianie
u - aktualizacja
d - usuwanie
r - reorganizacja pliku
```

argumenty dla poszczególnych instrukcji to

```
wstawianie - index promień wysokość
aktualizacja - index promień wysokość
usuwanie - index
reorganizacja pliku - brak
```

## 2 Sposób prezentacji wyników działania programu

### 2.1 Uruchomienie programu

W celu uruchomienia programu należy wykonać poniższa komende w folderze, gdzie znajduje sie plik main.py

```
python main.py
```

### 2.2 Opcje generowania rekordów

Istnieja 2 tryby pracy programu - interaktywny i tryb ładowania danych z pamieci. Aby rozpocząć w trybie interaktywnym należy kliknać "i"po starcie programu, analogicznie "l"dla trybu ładowania.

### 2.2.1 Tryb interaktywny

W trybie interaktywnym dostepne sa komendy :

```
i - wstawianie
d - usuwanie
u - akutalizacja
r - reorganizacja
v - wizualizacja plików indeksowego (z zachowaniem struktury drzewa) i pliku z danymi
p - pokazanie danych posortowanych po indeksach
g - wstawienie dowolnej liczby rekordów o losowych kluczach
```

### 2.2.2 Tryb ładoawnia

Po wybraniu trybu ładowania załadowany zostaje plik testowy zgodny z podanym wcześniej formatem i po wykonaniu wszystkich poleceń zawartość plików jest wyświetlana.

### 2.2.3 Format wyświetlania b-drzewa

Każdy wezeł b-drzewa wyświetlany jest w formacie

```
{ [adres dziecka 0] wartość indeksu 0 [adres dziecka 1] .... }
```

Przykładowe b-drzewo o korzeniu pod adresem 112. Dzieci korzenia znajduja sie pod adresami 0, 56, 168

```
112 - {[0], 11, [56], 51, [168]}
0 - {1, 3, 4}
56 - {23, 24, 48}
168 - {65, 89, 111}
```

# 3 Eksperyment

Eksperyment polega na zbadaniu korelacji pomiedzy stopniem drzewa a ilościa operacji dyskowych potrzebnych do usuniecia / wstawienia rekordu oraz zbadaniu korelacji miedzy stopniem zapełnienia drzewa ($\alpha$), obliczanym jako średnia stopniów zapełnienia każdego z wezłów, a ilościa operacji dyskowych potrzebnych do wstawiania / usuniecia rekordu.

## 3.1 Przegieg eksperymentu

Eksperyment jest wykonywany poprzez wykonanie nastepujacej sekwencji operacji 1000 razy dla każdego n $n \in \{100, 200, 400, 800\}$ i dla każdego $d \in \{2, 4, 6, 8, 10\}$ :

1) konstrukcja drzewa o stopniu $d$ poprzez wstawienie do pustego drzewa $n$ losowych, nie powtarzajacych sie rekordów

2) wykonanie operacji wstawienia i pomiar operacji dyskowych

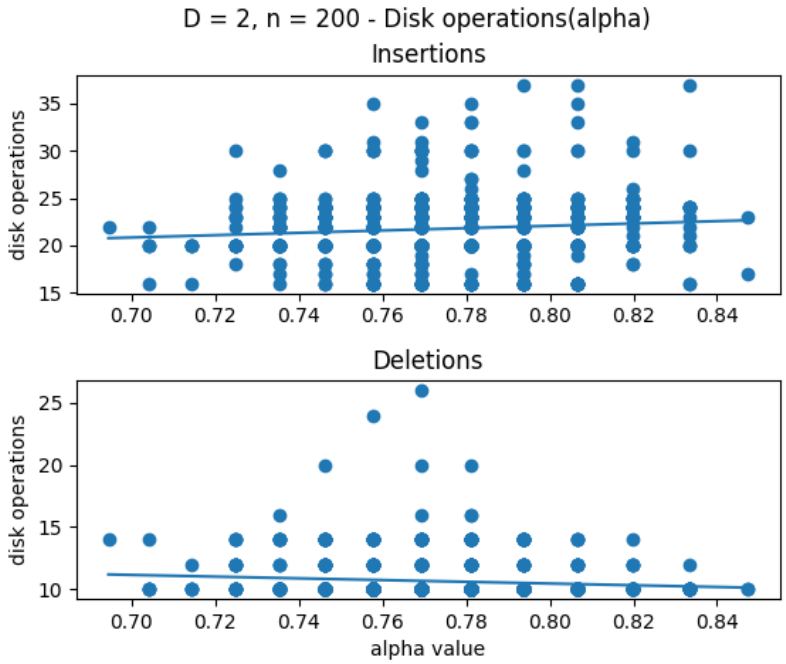
3) wykonanie operacji usuniecia losowego rekordu i pomiar operacji dyskowych

4) zapisanie wyniku wraz z obliczonym współczynnikiem alpha danego drzewa

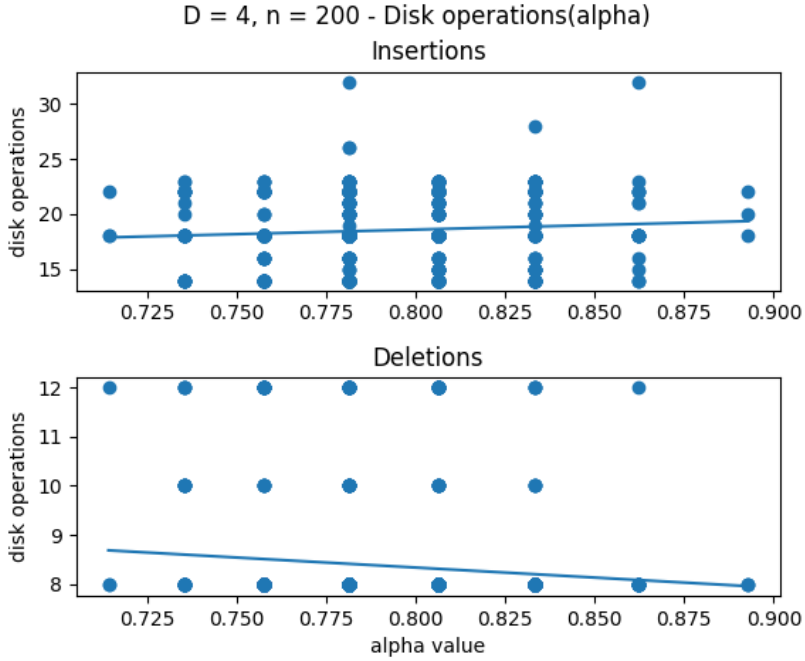
Finalnie do wyników zostaja dopasowane regresje liniowe w celu zbadania korelacji. Poniżej przedstawione sa wybrane wyniki badania korelacji miedzy współczynnikiem $\alpha$ a liczba operacji dyskowych dla $n \in \{200, 800\}$ i $d \in \{2, 4\}$ ( dla pozostałych n i d wyniki bardzo podobne)

**d = 2, n = 200**

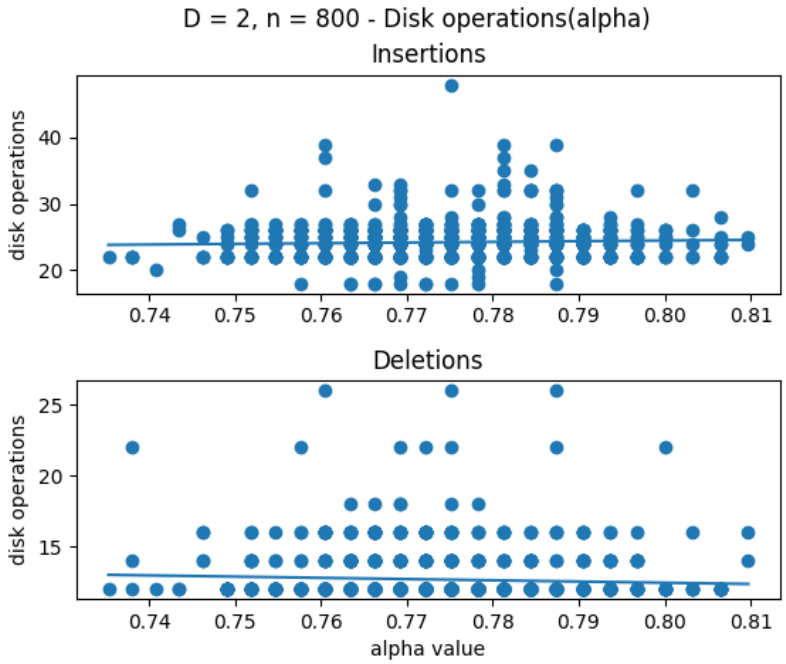
**dopasowana prosta dla wstawiania** $y = 12.33x + 12.20$ **i** $y = -6.94x + 16.00$ **dla usuwania**



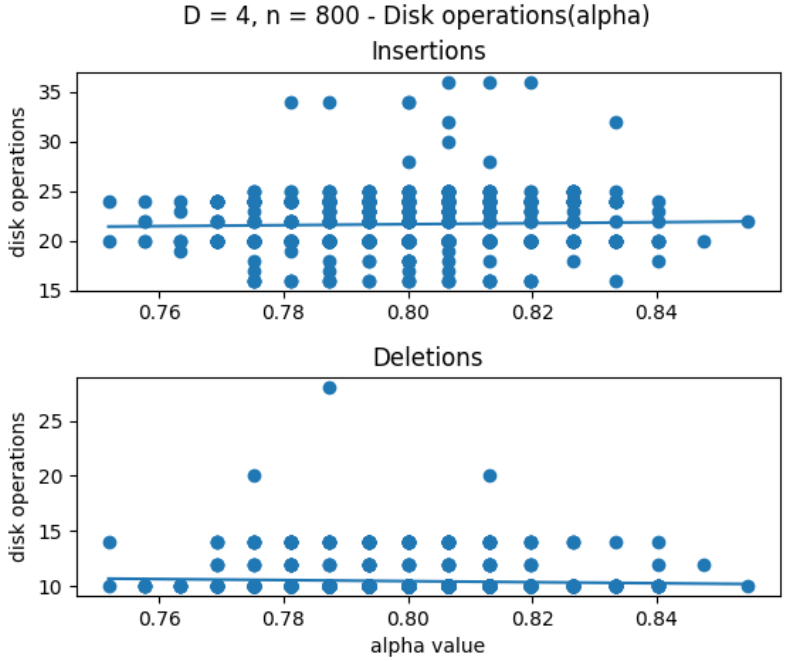

**d = 4, n = 200**
**dopasowana prosta dla wstawiania** $y = 8.25x + 11.99$ **i** $y = -4.08x + 11.60$ **dla usuwania**



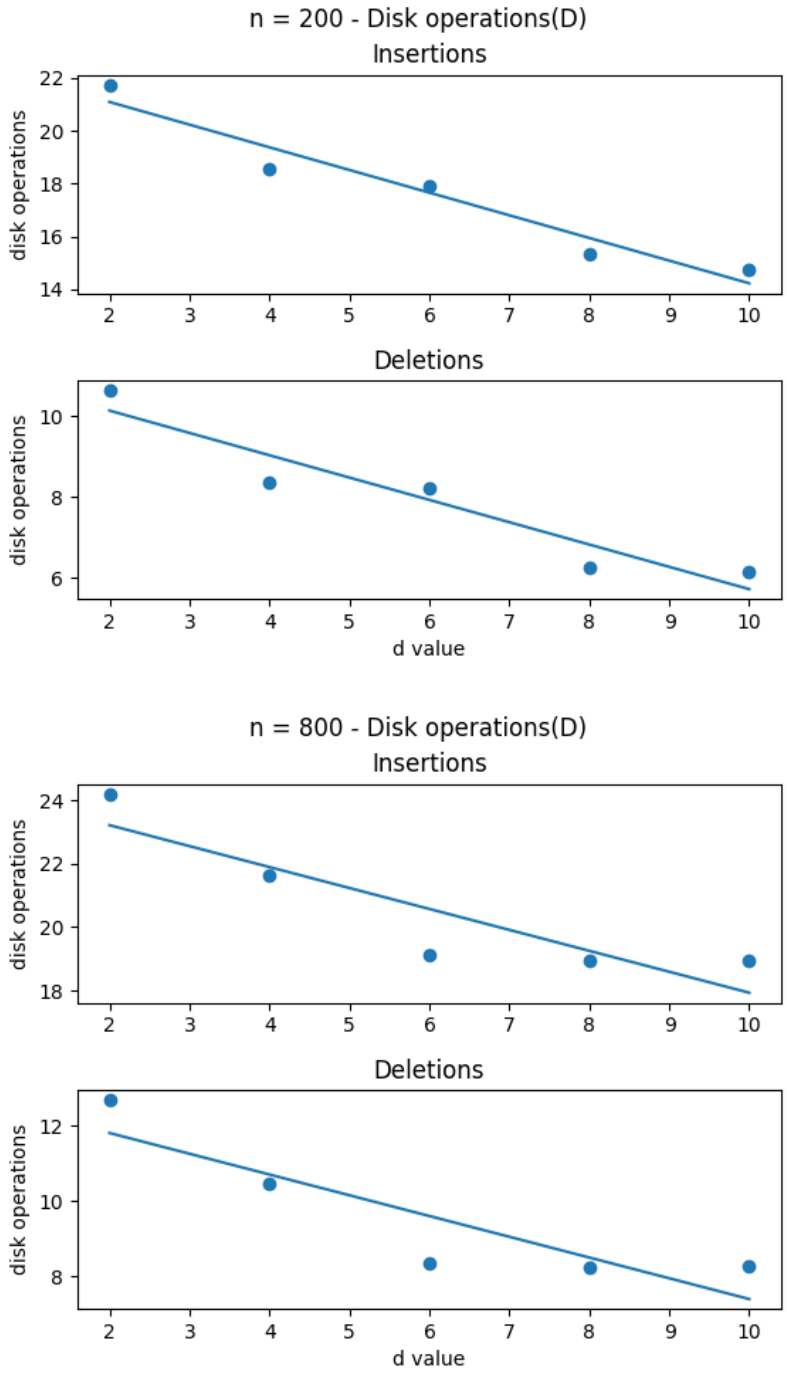

**d = 2, n = 800**
**dopasowana prosta dla wstawiania** $y = 10.06x + 16.40$ **i** $y = -8.67x + 19.39$ **dla usuwania**




**d = 4, n = 800**
**dopasowana prosta dla wstawiania** $y = 4.95x + 17.68$ **i** $y = -4.74x + 14.24$ **dla usuwania**




Poniżej przedstawiono wyniki badania korelacji miedzy stopniem drzewa a liczba operacji dyskowych dla wstawiania i usuwania dla $n \in \{200, 800\}$ ( dla pozostałych n wyniki bardzo podobne)




## 3.2 Wnioski

Podsumowujac, zauważalna jest silna korelacja ujemna miedzy liczba operacji dyskowych potrzebnych na usuwanie / dodawanie a stopniem drzewa, co jest spodziewane, gdyż złożoność tych operacji jest funkcja wysokości drzewa, drzewo o tej samej liczbie rekordów ale wyższym stopniu bedzie niższe. Zauważalna jest również korelacja dodania miedzy stopniem zapełnienia drzewa a złożonościa operacji wstawiania - im pełniejsze sa wezły tym cześciej bedzie nastepował podział, wiec operacja wstawiania bedzie wymagać wiecej operacji dyskowych. Istnieje również korelacja ujemna miedzy stopniem zapełnienia drzewa a ilościa operacji dyskowych potrzebnych do usuniecia rekordu - im pełniejsze wezły tym żadziej bedzie wystepować łaczenie dwóch wezłów po usunieciu.