8.01.2023r.

Struktury Baz Danych

Sprawozdanie z zadania 2.

Tomasz Piwowski, indeks 180171

1. Zastosowana metoda.

B-drzewo. Rekordy z przypisanymi kluczami/indeksami.

2. Opis implementacji.

Rozwiązania składa się z 2 głównych części:

- Symulator pamięci dyskowej.
- Implementacja B-drzewa z wykorzystaniem powyższego symulatora.

Całość jest wywołana z 2 niezależnych plików z metodą main: jeden jako standardowy (patrz: pkt 4.), drugi służący do przeprowadzania eksperymentów.

Symulator pamięci dyskowej jest obecny w postaci singletona, który dostępny jest z każdego miejsca w kodzie. Posiada pamięć cache zdolną do przechowania 1 strony z rekordami oraz potencjalnie nieograniczonej, ale efektywnie ograniczonej do wysokości drzewa i najbliższych sąsiadów niektórych z jego węzłów, listy stron reprezentującej poszczególne węzły B-drzewa.

Wielkość strony dyskowej jest wyrażona w bajtach. W przypadku stron z rekordami, na stronie zapisywane jest tylko tyle rekordów ile się zmieści w całości, reszta bajtów jest nieużywana. Dla stron z węzłami — na końcu każdej strony znajduje się adres do kolejnej strony reprezentującej naturalną kontynuacje strony, w ten sposób 1 węzeł może mieścić się na więcej niż 1 stronie.

Strona z rekordami jest zapisywana tylko gdy potrzebny jest dostęp do innej strony z rekordami.

Strony z węzłami są zapisywane po każdej operacji zapisu/usunięcia.

Implementacja B-drzewa jest wyrażona w postaci standardowej metody z klasą węzła B-drzewa oraz klasą nadrzędną B-drzewa, która zawiera jego korzeń oraz spełnia wszystkie wymagane funkcjonalności.

Węzeł B-drzewa wykorzystuje listę o określonej maksymalnej długości, której elementy zawierają: beforeNode (int; adres pamięci do potomka z kluczami mniejszymi niż klucz z tego elementu), klucz oraz adres pamięci z rekordem dla tego klucza. Ostatni element zawiera jedynie beforeNode, który w tym przypadku interpretowany jest jako afterNode (analogicznie: adres pamięci do potomka z kluczami większymi niż obecne w węźle klucze), pozostałe elementy (klucz i adres do rekordu) są niezdefiniowane, a ich wartość niesprecyzowana (zazwyczaj 0, 0, ale może zawierać cokolwiek).

Węzeł B-drzewa przy zapisywaniu strony do pamięci zapisuje jedynie:

- Ilość elementów w liści,
- Poszczególne elementy w liście,
- afterNode,
- adres do strony z kontynuacją (jeśli trzeba).

Przy odczycie zawsze w pierwszej kolejności sprawdza cache, a potem odczytuje z pliku. Zapisywane do pliku węzły sobie dopiero na koniec operacji wstawiania/usuwania z klasy głównej B-drzewa.

Operacje dodawania/usuwania węzła:

Dodawanie — pobiera klucz i wyszukuje miejsce dla niego w węźle (wyjątek: klucz już obecny w węźle). Operacje kompensacji oraz split są wykonywane na atrapie relacji: dzieckoLewe-rodzic-dzieckoPrawe, gdzie rodzica symbolizuje pojedynczy element z listy rodzica. Następnie do takiej atrapy dodawany jest element oraz jest ona dołączana do rodzica (w przypadku kompensacji element rodzica zostaje nadpisany przez element atrapy)

Usuwanie — pobiera klucz i wyszukuje go w węźle (jeśli klucz nie istnieje w węźle, wraca nie robiąc nic). Jeśli element jest liściem, zamienia klucz i adres jego rekordu z następnikiem elementu. Następnie usuwa element z liścia.

Usuwanie elementu z liścia (kompensacja oraz merge) również dzieje się na bazie atrapy, która następnie jest aplikowana do rodzina (nadpisywany jest element rodzica lub usuwany w przypadku operacji merge).

Węzeł B-drzewa posiada operacje wypisania w postaci drzewiastej oraz funkcję forEach dla każdego elementu z jego listy i list jego potomków. Obie z tych funkcji nie zapisują węzłów do cache, gdyż nie jest to w tym wypadku pożądane zachowanie.

Dodatkowo, każdy węzeł zapamiętuje adres strony dyskowej, z której został zapisany, a cache przeglądane jest tylko w przypadku próby znalezienie lewego lub prawego rodzeństwa.

3. Specyfikacja formatu pliku tekstowego.

Plik tekstowy przeznaczony do automatyzacji budowania b-drzewa wykorzystuje 4 rodzaje komend:

- add [klucz] [wartość 1] [wartość 2] [wartość 3] [wartość 4] [wartość 5]

- del [klucz]
- show
- records

Gdzie wyrażenie objęte kwadratowym nawiasem oznacza dowolną liczbę.

Każda komenda powinna być oddzielona znakiem nowej linii. Komendy wykonywane są sekwencyjnie, zgodnie z kolejnością podaną w pliku.

Add — dodaje rekord do drzewa (rekord składa się z klucza i 5 wartości).

Add z istniejącym kluczem — powoduje nadpisanie istniejące klucza w miejscu (bez usuwania i wstawiania go na nowo).

Del — usuwa rekord o zadanym kluczu z drzewa.

Show — wyświetla drzewo z zaznaczeniem poszczególnych węzłów i zależności między nimi (rodzic, syn, rodzeństwo). Nie wyświetla pełnych rekordów, a jedynie ich klucze.

Records — wyświetla wszystkie rekordy B drzewa w kolejności rosnącej po kluczach.

Dodatkowo jest możliwość wprowadzania komentarzy, które zaczynać muszą się znakiem # i spacją oraz kończyć się analogicznie spacją i znakiem #. Treść komentarza jest wypisywana na konsoli podczas wykonywania programu.

4. Sposób prezentacji wyników działania programu.

Uruchomienie programu (domyślnie w trybie interaktywnym). Po wprowadzeniu odpowiedniej komendy (jak w pkt. wyżej) i wykonaniu odpowiadającej jej funkcji, zostaną wyświetlone parametry wykonania takie jak:

- Parametr D,
- Ilość rekordów w drzewie,
- Wysokość drzewa,
- Ilość dokonanych zapisów stron na dysku,
- Ilość dokonanych odczytów stron z dysku.

Program kończy się po wprowadzeniu komendy „quit”.

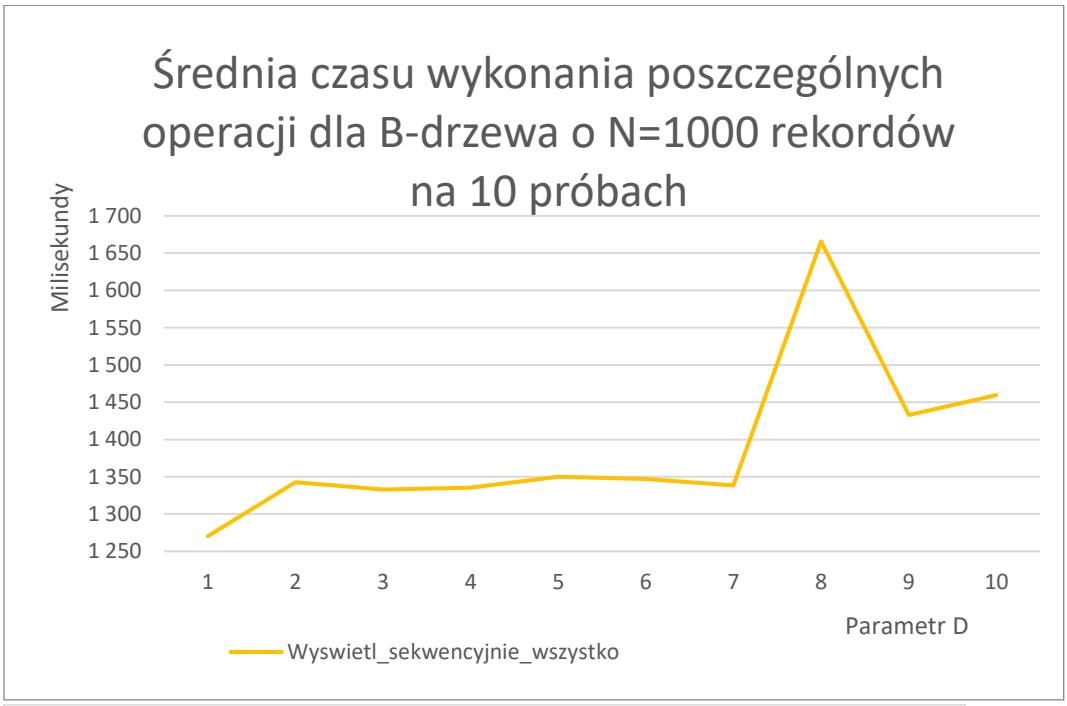
W celu użycia programu z wykorzystanie pliku z komendami do wywołania należy podać ścieżkę do pliku w polu argumentów wywołania. Plik zostanie załadowany, a dodatkowo przed każdą wykonaną komendą zostanie podany numer linii z pliku, z którego pochodzi komenda. Po wykonaniu ostatniej komendy z pliku, program przechodzi w tryb interaktywny.

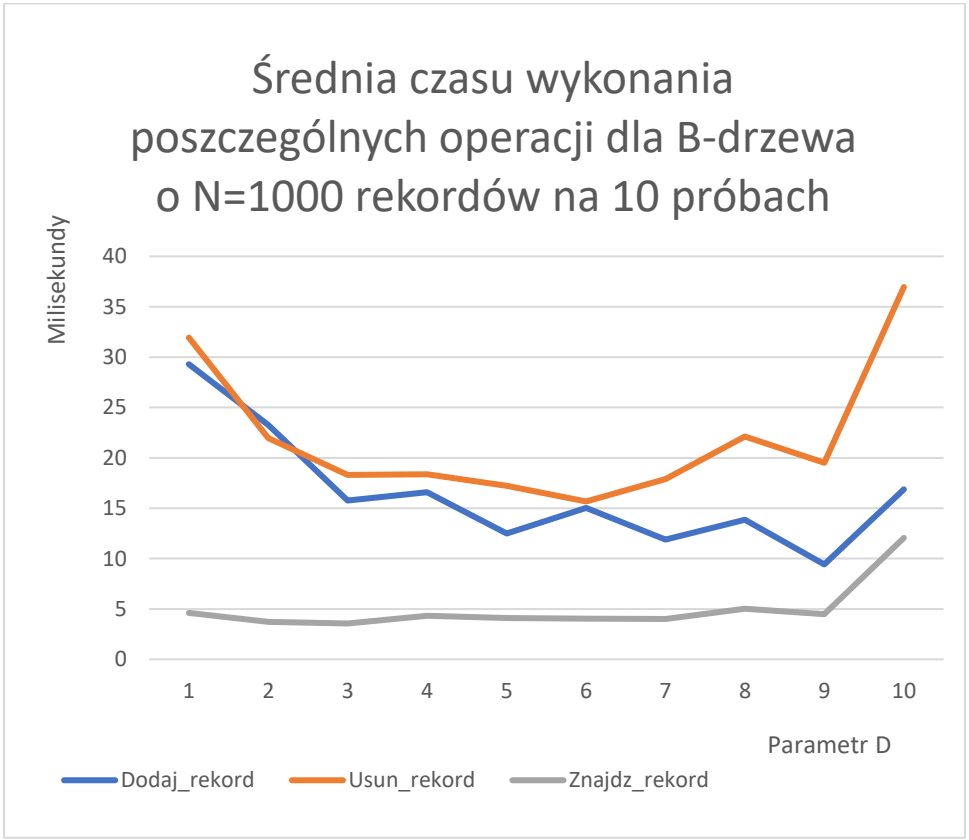
5. Eksperyment.

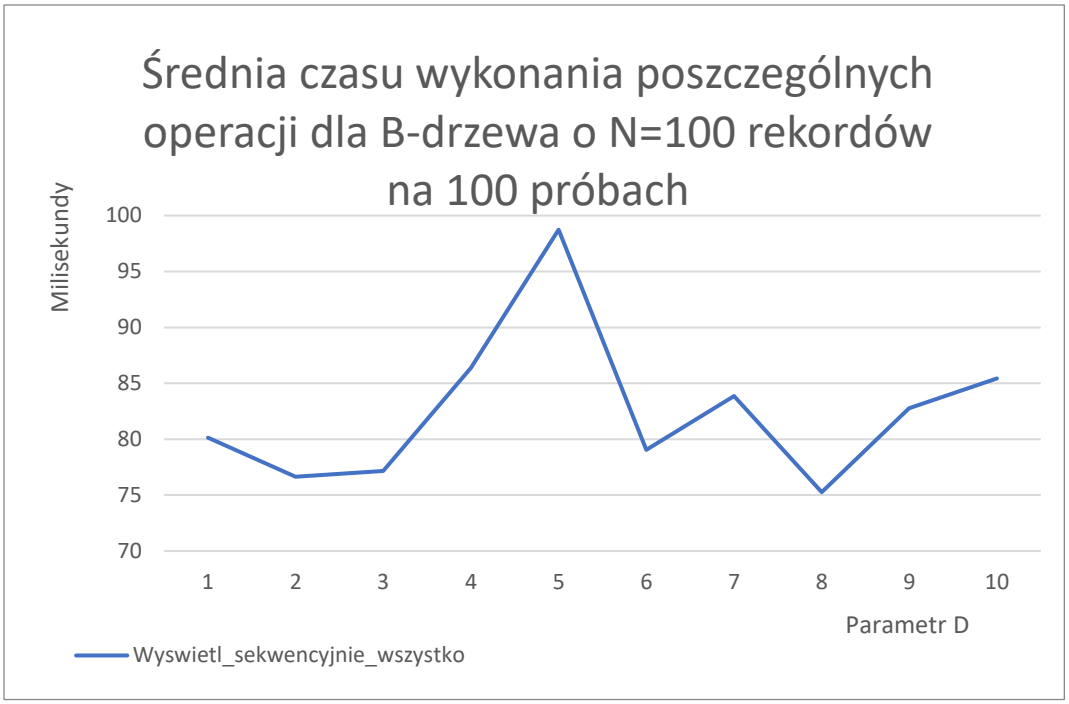
W ramach eksperymentu została sprawdzona doświadczalnie złożoność czasowa operacji na B-drzewie. Przed każdą operacją wygenerowane zostało losowe B-drzewo poprzez dodanie N

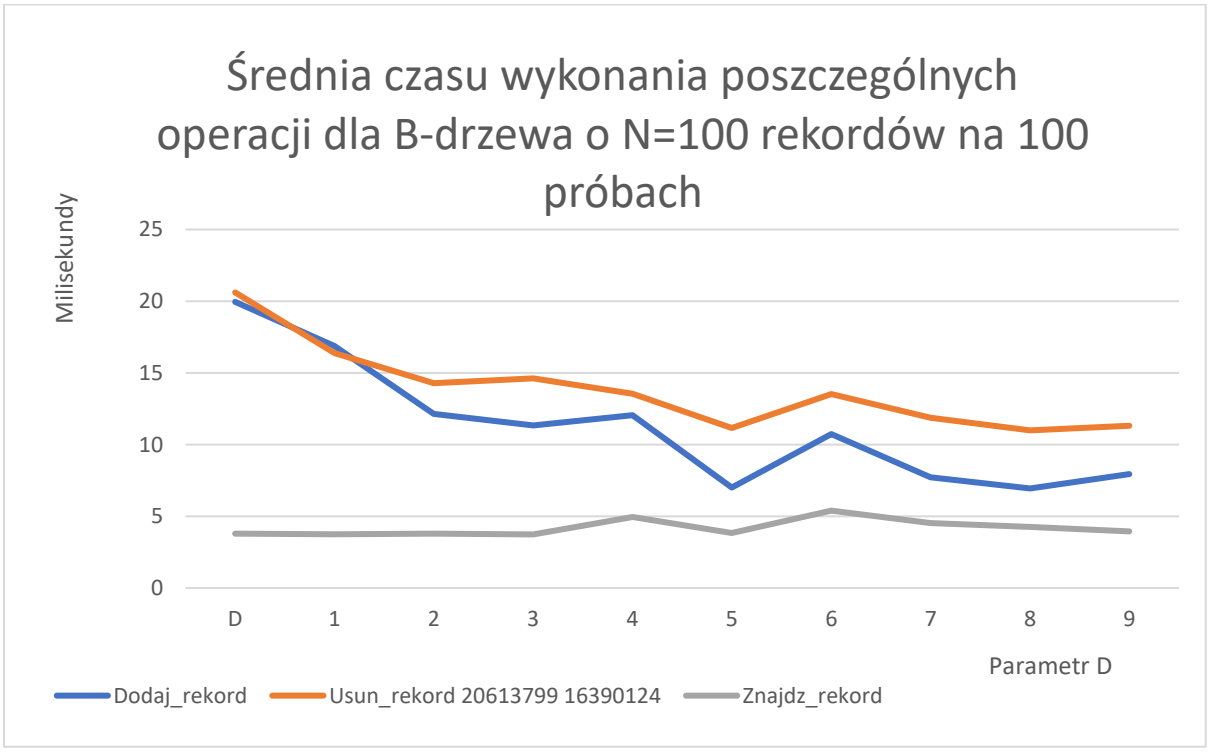
rekordów o losowych kluczach z przedziału 0-64 tyś. Następnie na podstawie wszystkich prób wyliczona została średnia czasu wykonania.

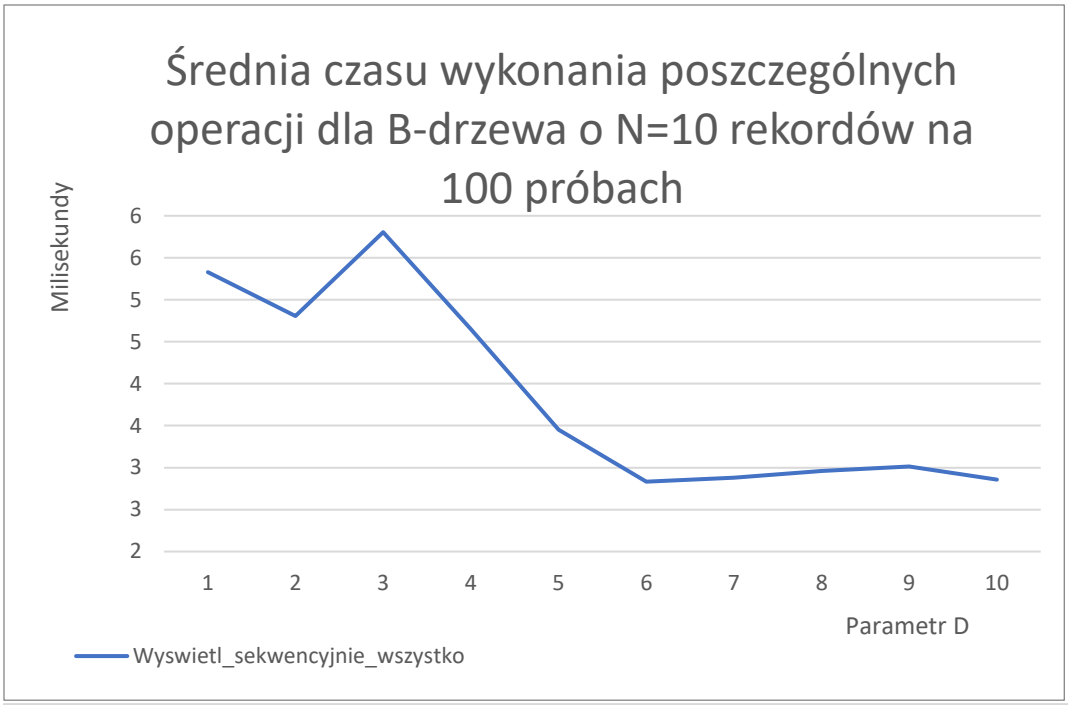
Oto wyniki eksperymentu:

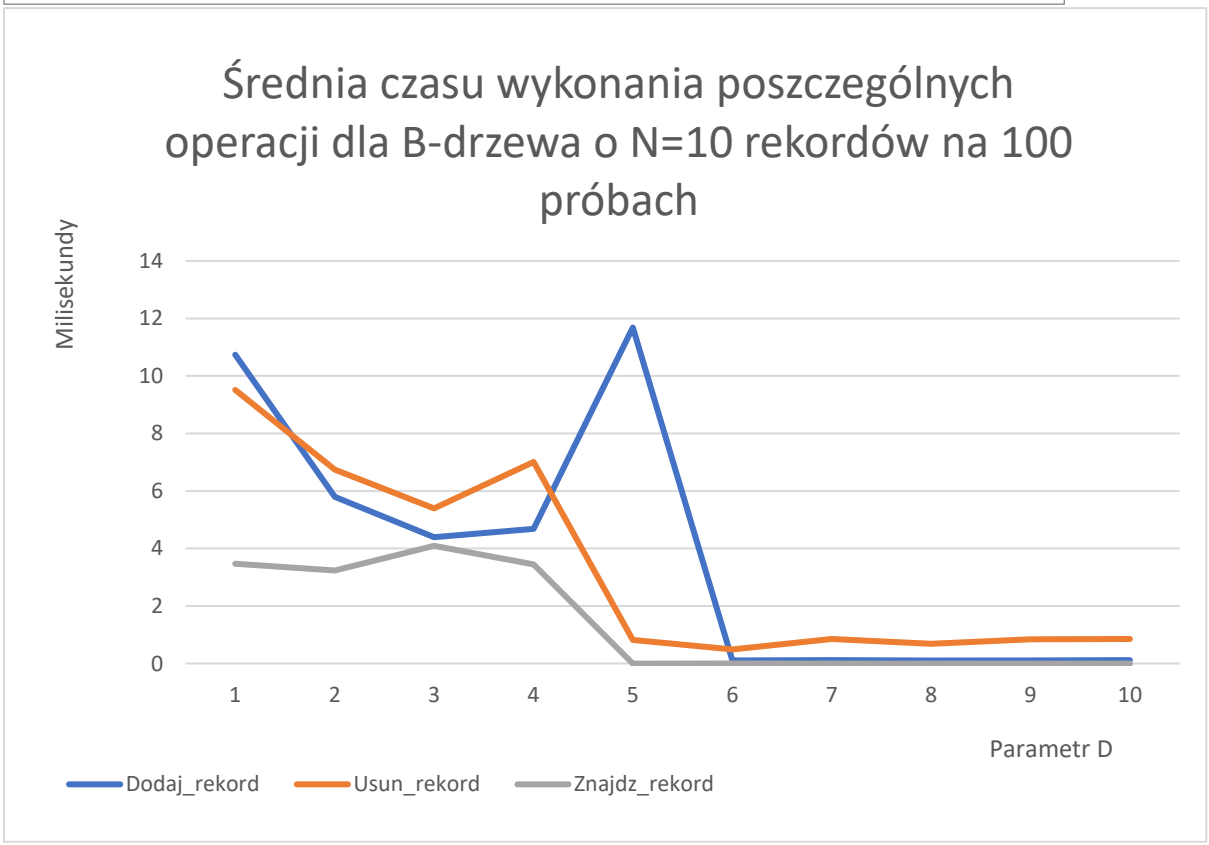


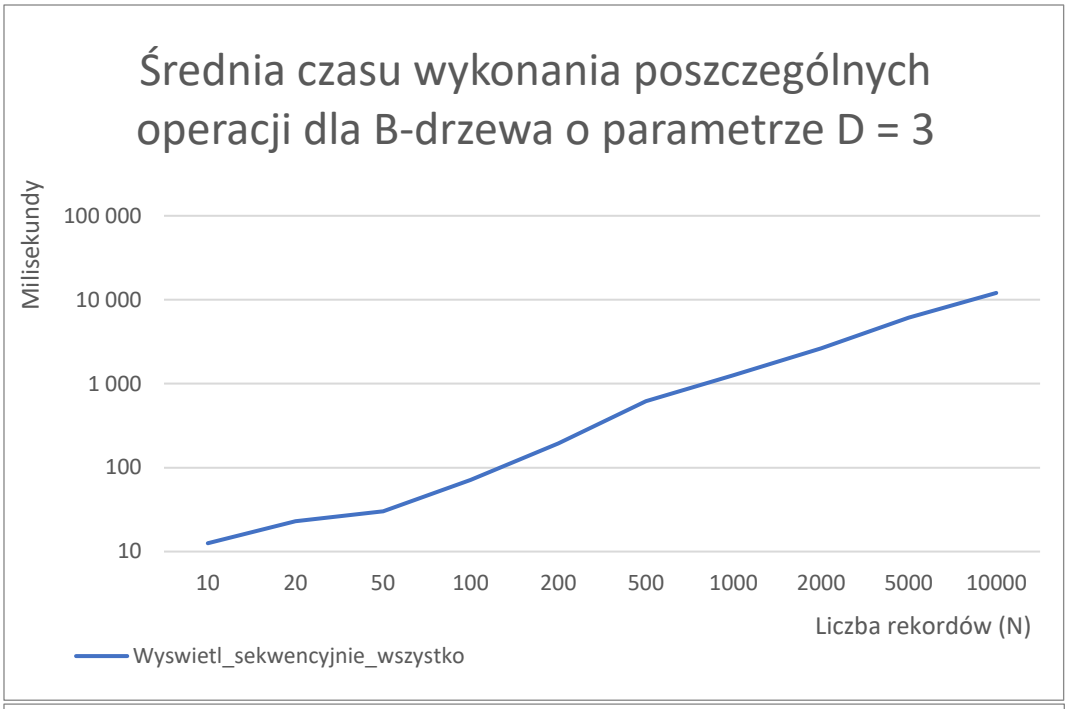


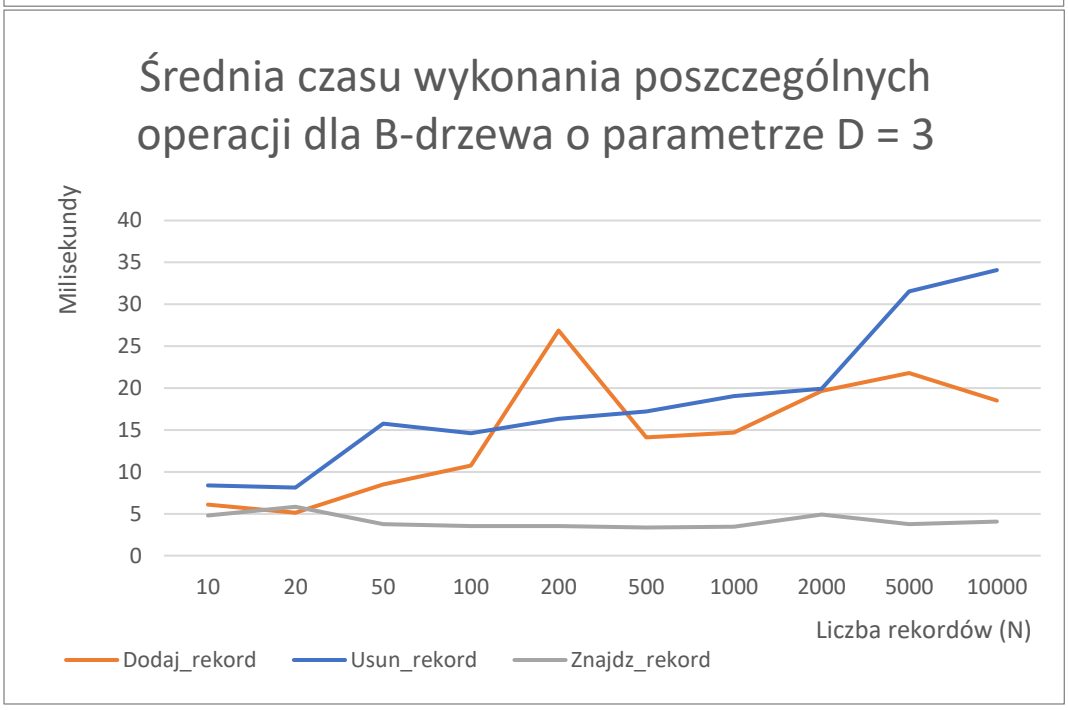

Średnia czasu wykonania poszczególnych operacji dla B-drzewa o N=100 rekordów na 100 próbach
Milisekundy
100
95
90
85
80
75
70
1
2
3
4
5
6
7
8
9
10
Parametr D
Wyswietl_sekwencyjnie_wszystko


Średnia czasu wykonania poszczególnych operacji dla B-drzewa o N=100 rekordów na 100 próbach
Milisekundy
25
20
15
10
5
0
D
1
2
3
4
5
6
7
8
9
Parametr D
Dodaj_rekord
Usun_rekord 20613799 16390124
Znajdz_rekord


Średnia czasu wykonania poszczególnych operacji dla B-drzewa o N=10 rekordów na 100 próbach
Milisekundy
6
6
5
5
4
4
3
3
2
1
2
3
4
5
6
7
8
9
10
Parametr D
Wyswietl_sekwencyjnie_wszystko


Średnia czasu wykonania poszczególnych operacji dla B-drzewa o N=10 rekordów na 100 próbach
Milisekundy
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9
10
Parametr D
Dodaj_rekord
Usun_rekord
Znajdz_rekord





Na podstawie przytoczonych wyników można stwierdzić, że istnieje optymalna wartość parametru D, dla której złożoność operacji jest najlepsza. Powyżej tej wartości sekwencyjne odczytywanie pliku dla dużych N pogarsza się, a przy bardzo małych wartościach większość operacji nie jest w pełni optymalna.

Eksperyment wskazuje, że sekwencyjne wyświetlanie rekordów B-drzewa jest O(n) (złożoność liniowa). Natomiast pozostałem operacje takie jak: dodawanie, usuwanie rekordów są O(logn). Znajdywanie rekordów wg eksperymentu pozostaje stałe czasowe (O(1)).

N=100	Dodaj_rekord		Usun_rekord		Znajdz_rekord		Wyswietl_sekwencyjnie_wszystko	
D	Odczyty	Zapisy	Odczyty	Zapisy	Odczyty	Zapisy	Odczyty	Zapisy
1	5	5	5	5	4	0	110	1
2	3	4	3	3	2	0	78	1
3	2	2	2	2	2	0	69	1
4	2	2	2	2	2	0	63	1
5	1	2	1	1	1	0	59	1
6	1	1	1	1	1	0	57	1
7	1	1	1	1	1	0	56	1
8	1	1	1	1	1	0	54	1
9	1	1	1	1	1	0	55	1
10	1	1	1	1	1	0	53	1

Parametr D wpływa natomiast na liczbę zapisów/odczytów przy operacjach. Jednak przy pewnej wartości dalsze jego zwiększanie nie powoduje już znacznego wzrostu efektywności. Małe D oznacza potencjalnie większą ilość operacji dyskowych.

D=3	Dodaj_rekord		Usun_rekord		Znajdz_rekord		Wyswietl_sekwencyjnie_wszystko	
N	Odczyty	Zapisy	Odczyty	Zapisy	Odczyty	Zapisy	Odczyty	Zapisy
10	1	1	1	1	1	0	2	0
20	1	1	1	1	1	0	10	1
50	2	2	3	3	2	0	26	1
100	3	3	3	3	2	0	63	1
200	5	7	4	3	2	0	182	1
500	4	4	4	4	3	0	541	1
1000	4	4	5	5	3	0	1123	1
2000	5	5	5	5	4	0	2315	1
5000	5	5	7	7	4	0	5775	1
10000	5	5	6	6	5	0	11115	1

Zgodnie z przewidywaniami zapis, usunięcie oraz znalezienie rekordu wymaga O(logn) operacji dyskowych, co równa się w przybliżeniu wysokości drzewa (trzeba wczytać/zapisać 1 ścieżkę drzewa).

Z kolei wyświetlenie sekwencyjne nie jest już zbyt efektywne. Losowo dodawane rekordy nie są umieszczone na stronach dyskowych tak, by na jednej stronie znajdowały się kolejne rekordy (wg. ustalonego porządku). W związku z tym Liczba odczytów staje się proporcjonalna do ilości rekordów (w przypadku każdego rekordu należy odczytać nową stronę dyskową + narzut w postaci stron zawierających węzły B-drzewa).