

# Struktury baz danych

## Projekt 2 – B-drzewo

Mateusz Kowalczyk, s188717

### 1. Wprowadzenie

Projekt miał na celu implementację oraz testowanie struktury indeksowej – B-drzewa.

Program w pamięci operacyjnej posiada 2 stałe bufor – 1 przeznaczony do odczytu i zapisu z pliku głównego oraz 1 do tychże operacji dotyczących pliku indeksowego.

Każda strona B-drzewa, oprócz kluczy i indeksów rekordów, posiada aktualną liczbę kluczy oraz indeks strony rodzica. Liczba –1 używana jest jako NIL.

Rekordem jest tablica 28 bajtów następującej postaci:

[8 B: obwód pięciokąta][4 B: bok pięciokąta][4 B: bok pięciokąta]...[4 B: bok pięciokąta].

Zaimplementowano następujące operacje:

- wstawianie rekordu:
  - operacje rozbicia (*split*) powoduje przeniesienie rekordów o mniejszym kluczu do nowej strony na tym samym poziomie – nowy „brat” tworzy się zawsze po lewej stronie; na potrzeby operowania na dwóch buforach jednocześnie w pamięci tworzony jest dodatkowy tymczasowy bufor używany tylko do zakończenia tej operacji;
  - operację kompensacji (*compensation*) wykonuje się w prawo tylko wtedy, gdy nie można wykonać jej w lewo – najpierw sprawdzany jest warunek dotyczący liczby rekordów w lewym „bracie”; w celu zapisania w pamięci operacyjnej wszystkich rekordów przenoszonych do „brata” wykorzystuje się nowy tymczasowy bufor w postaci listy, której elementami są czwórki zawierające: klucz rekordu, indeks rekordu, indeks lewego dziecka, indeks prawego dziecka;
- wyszukiwanie jednego rekordu po kluczu:
  - każdemu rekordowi przypisany jest jego indeks stanowiący pozycję w pliku głównym; ze względu na stały rozmiar rekordu można obliczyć, na której stronie i na której pozycji w swojej stronie się on znajduje;
- wypisanie wszystkich rekordów w kolejności rosnącego klucza:
  - ze względu na wyłącznie jeden dostępny bufor przypisany do pliku indeksowego operacja ta jest kosztowna i wymaga wielokrotnego odczytywania tych samych stron B-drzewa;
- wyświetlenie zawartości pliku głównego i indeksowego w całości:
  - operacja ta wyświetla w sposób czytelny dla człowieka zawartości plików, prezentując ją w podziale na strony.

Po każdej z powyższych operacji wyświetlona zostaje liczba odczytów i zapisów na dysk (tę funkcję można wyłączyć).

```

Main menu:
0: Toggle menu output mode (set to: True)
1: Toggle records in the file number showing (set to: False)
2: Create new file
3: Add records
4: Get record by key
5: Get all records (sorted)
6: Show main and index file content
7: Exit
> 3

Records adding menu:
- records number: 20

- record array values: 23 7 5 6 21
disk reads: 0
disk writes: 1
- record array values: 6 5 2 2 8
disk reads: 0
disk writes: 1

```

## 2. Interakcja z programem

Z programu można korzystać w sposób interaktywny lub wczytując dane wejściowe z pliku testowego. W pliku testowym umieszcza się w osobnych liniach numery poleceń i wprowadzane dane w ten sam sposób, w który wprowadzałoby się je do interaktywnie działającego programu. Najważniejszych poleceń używa się następująco:

- stworzenie pliku:
  - interaktywnie:

```

Main menu:
0: Toggle menu output mode (set to: True)
1: Toggle records in the file number showing (set to: False)
2: Create new file
3: Add records
4: Get record by key
5: Get all records (sorted)
6: Show main and index file content
7: Exit
> 2

File creating menu
- filename: file0

```

- w pliku testowym:
 

```

2
file0

```
- dodanie rekordów:
  - interaktywnie:

```

Main menu:
0: Toggle menu output mode (set to: True)
1: Toggle records in the file number showing (set to: False)
2: Create new file
3: Add records
4: Get record by key
5: Get all records (sorted)
6: Show main and index file content
7: Exit
> 3

Records adding menu:
- records number: 5

- record array values: 1 1 1 1 1
disk reads: 0
disk writes: 1
- record array values: 2 2 2 2 2
disk reads: 0
disk writes: 1
- record array values: 3 3 3 3 3
disk reads: 0
disk writes: 1
- record array values: 4 4 4 4 4
disk reads: 0
disk writes: 1
- record array values: 5 5 5 5 5
disk reads: 4
disk writes: 7

```

- w pliku testowym:
  - 3
  - 5
  - 1 1 1 1 1
  - 2 2 2 2 2
  - 3 3 3 3 3
  - 4 4 4 4 4
  - 5 5 5 5 5
- wyszukiwanie jednego rekordu po kluczu:
  - interaktywnie:

```

Main menu:
0: Toggle menu output mode (set to: True)
1: Toggle records in the file number showing (set to: False)
2: Create new file
3: Add records
4: Get record by key
5: Get all records (sorted)
6: Show main and index file content
7: Exit
> 4

Record reading by index menu:
- key: 20
[key: 20] 5 5 5 5
disk reads: 1
disk writes: 0

```

- w pliku testowym:
  - 4
  - 20
- wypisanie wszystkich rekordów w kolejności rosnącego klucza:
  - interaktywnie:

```

Main menu:
0: Toggle menu output mode (set to: True)
1: Toggle records in the file number showing (set to: False)
2: Create new file
3: Add records
4: Get record by key
5: Get all records (sorted)
6: Show main and index file content
7: Exit
> 5
[key: 4] 1 1 1 1
[key: 8] 2 2 2 2
[key: 12] 3 3 3 3
[key: 16] 4 4 4 4
[key: 20] 5 5 5 5
disk reads: 7
disk writes: 0

```

- w pliku testowym:
  - 5
- wyświetlenie zawartości pliku głównego i indeksowego w całości:
  - interaktywnie:

```
Main menu:
0: Toggle menu output mode (set to: True)
1: Toggle records in the file number showing (set to: False)
2: Create new file
3: Add records
4: Get record by key
5: Get all records (sorted)
6: Show main and index file content
7: Exit
> 6

[MAIN FILE][page: 0]
[record: 0][key: 4] 1 1 1 1
[record: 1][key: 8] 2 2 2 2
[record: 2][key: 12] 3 3 3 3
[record: 3][key: 16] 4 4 4 4

[MAIN FILE][page: 1]
[record: 4][key: 20] 5 5 5 5
[record: 5][key: 0] 0 0 0 0
[record: 6][key: 0] 0 0 0 0
[record: 7][key: 0] 0 0 0 0
```

```

[INDEX FILE][page: 0]
[keys count: 2][parent page: 2]
[child page: -1]
[key: 16][record: 3]
[child page: -1]
[key: 20][record: 4]
[child page: -1]
[key: 0][record: 0]
[child page: 0]
[key: 0][record: 0]
[child page: 0]

[INDEX FILE][page: 1]
[keys count: 2][parent page: 2]
[child page: -1]
[key: 4][record: 0]
[child page: -1]
[key: 8][record: 1]
[child page: -1]
[key: 0][record: 0]
[child page: 0]
[key: 0][record: 0]
[child page: 0]

[INDEX FILE][page: 2]
[keys count: 1][parent page: -1]
[child page: 1]
[key: 12][record: 2]
[child page: 0]
[key: 0][record: 0]
[child page: 0]
[key: 0][record: 0]
[child page: 0]
[key: 0][record: 0]
[child page: 0]
disk reads: 5
disk writes: 0

```

o w pliku testowym:

6

- wyjście z programu:
  - o interaktywnie:

```
Main menu:
0: Toggle menu output mode (set to: True)
1: Toggle records in the file number showing (set to: False)
2: Create new file
3: Add records
4: Get record by key
5: Get all records (sorted)
6: Show main and index file content
7: Exit
> 7
```

o w pliku testowym (należy obowiązkowo umieścić w ostatniej linii):

7

Przykładowa zawartość pliku testowego:

```
2
file0
3
5
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
4
3
5
6
7
```

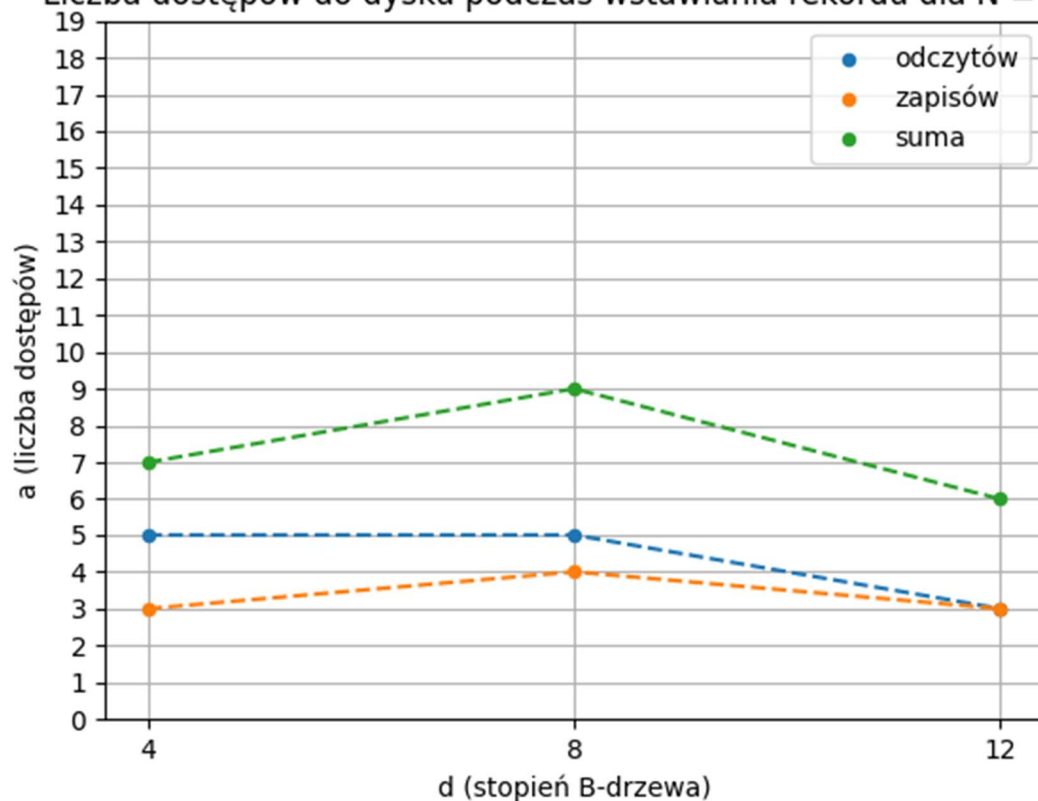
### 3. Eksperymenty

W każdym eksperymencie minimalną wartością klucza było 5, a maksymalną  $10^6$ .

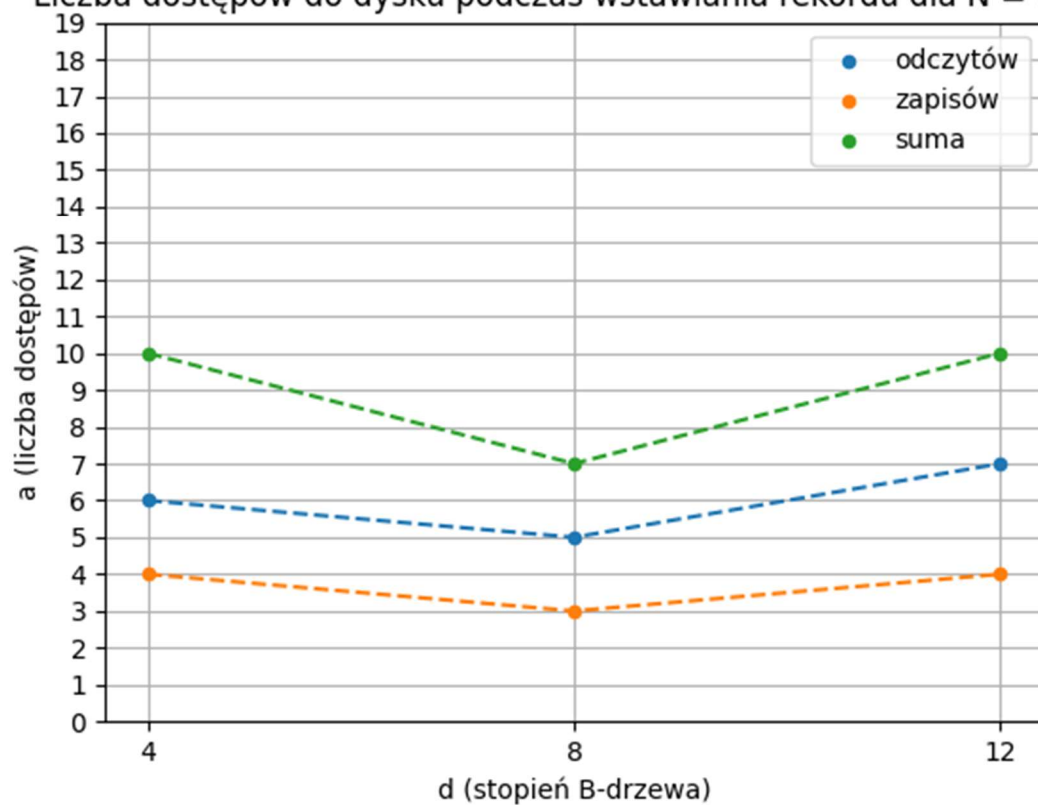
a) Średnia liczbaostępów do dysku podczas wstawiania nowego rekordu.

Wybrano 3 różne stopnie B-drzewa  $d$  i 3 różne liczby  $N$  już dodanych rekordów. Przygotowano po 5 plików testowych z poleceniami dodania każdej liczby  $N$  losowych rekordów oraz 1 dodatkowego (nowego) rekordu. Dla każdej pary  $(d, N)$  i dla każdego pliku testowego odczytano liczbę dostępów do dysku przy dodawaniu odpowiedniego nowego rekordu. Dla każdej pary  $(d, N)$  uśredniono wynik. Dla każdego stopnia  $d$  liczbę rekordów przypadających na stronę pliku głównego ustawiano na  $2d$ . Rezultat prezentuje się następująco:

Liczba dostępuów do dysku podczas wstawiania rekordu dla  $N = 100$

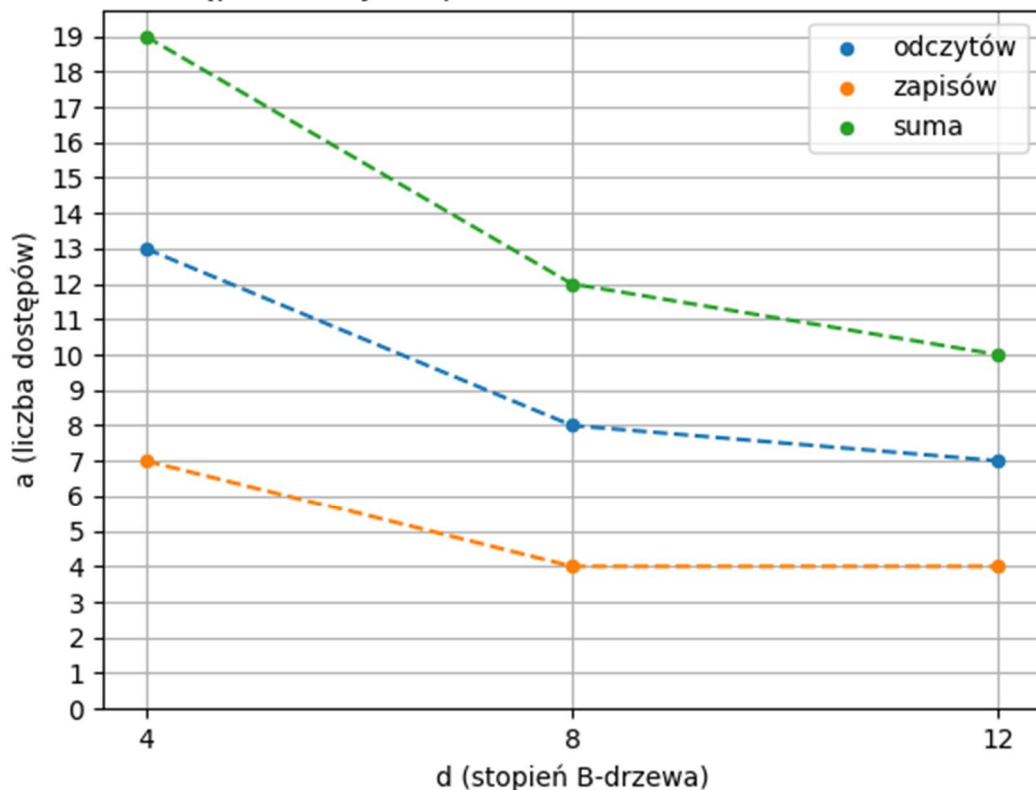


Liczba dostępuów do dysku podczas wstawiania rekordu dla  $N = 1000$





Liczba dostępów do dysku podczas wstawiania rekordu dla  $N = 10000$

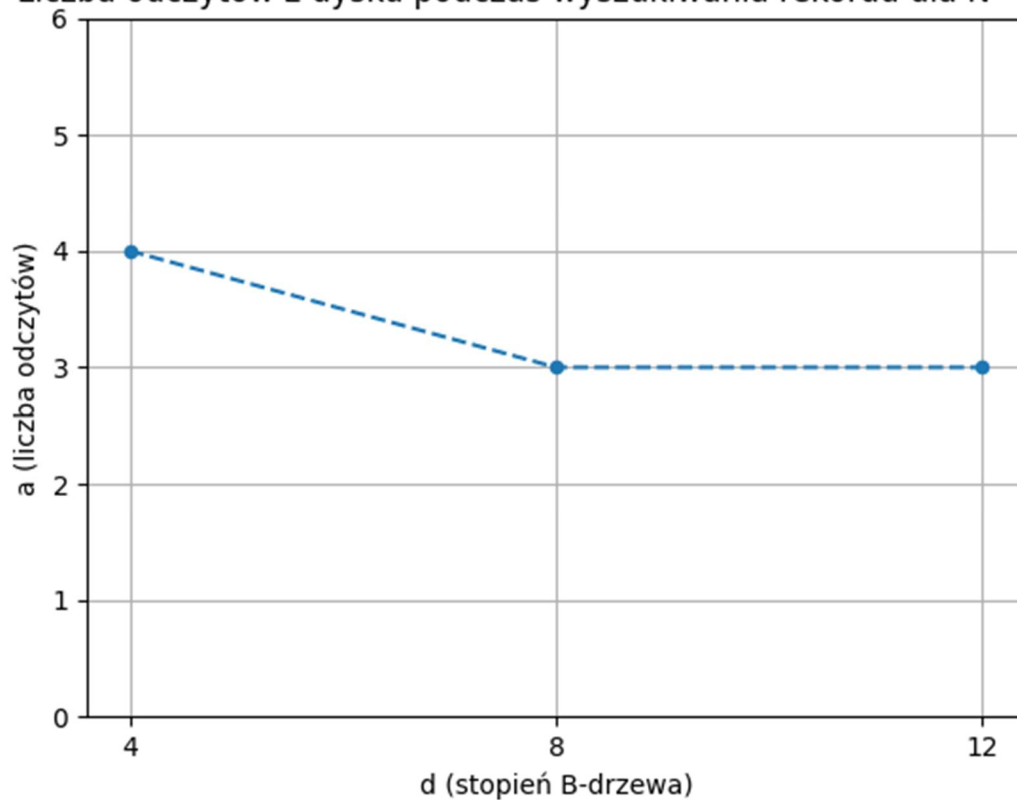


Na wykresach widać, iż wzrost stopnia B-drzewa nie zawsze powoduje zmniejszenie liczb dostępów do dysku. Można wnioskować, iż znaczenie ma tu nie tylko liczba obecnych już rekordów, ale w dużym stopniu wpływają również wartości klucza rekordu wstawianego i rekordów już obecnych. Wartość  $a$  można by zmniejszyć kosztem większej liczby buforów w pamięci operacyjnej. Wówczas operacje, które wymagają odczytu z kilku stron (rozbicie, kompensacja) mogłyby pobrać ich więcej do pamięci zamiast przełączać się między nimi, wielokrotnie odczytując tę samą stronę.

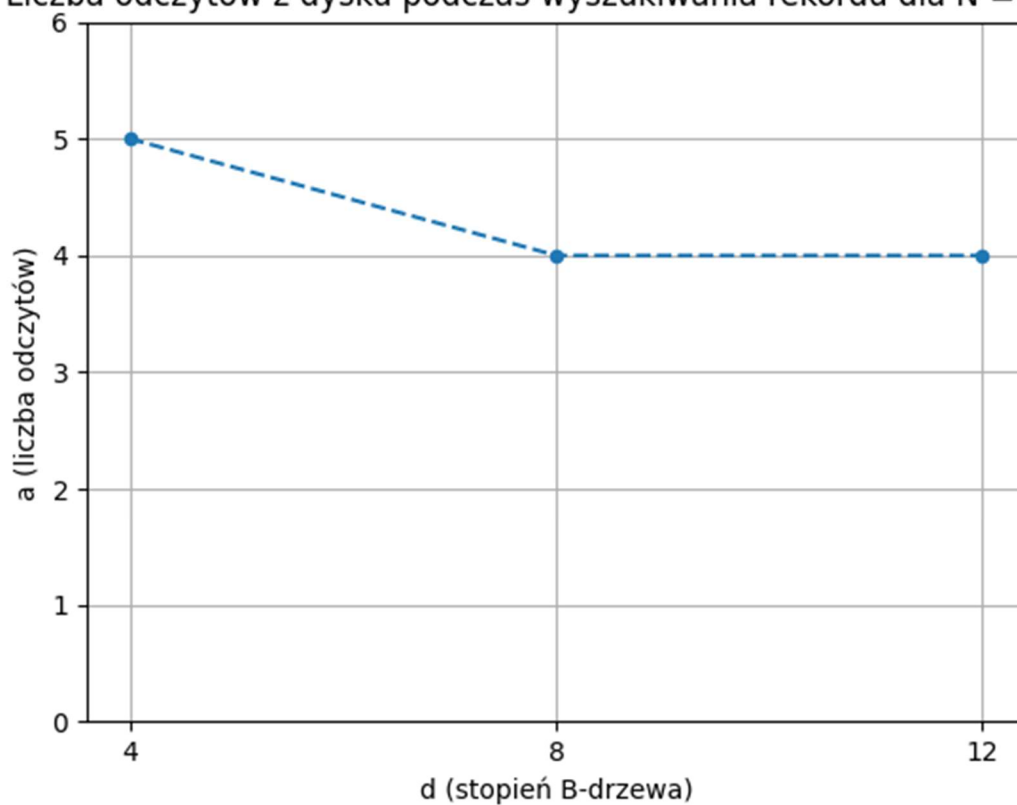
b) Średnia liczba dostępów do dysku podczas odczytu rekordu.

Wybrano 3 różne stopnie B-drzewa  $d$  i 3 różne liczby  $N$  obecnych w pliku rekordów. Przygotowano po 5 plików testowych z poleceniami dodania każdej liczby  $N$  losowych rekordów oraz wyszukania losowego spośród nich. Dla każdej pary  $(d, N)$  i dla każdego pliku testowego odczytano liczbę dostępów do dysku przy wyszukiwaniu odpowiedniego rekordu. Dla każdej pary  $(d, N)$  uśredniono wynik. Dla każdego stopnia  $d$  liczbę rekordów przypadających na stronę pliku głównego ustawiano na  $2d$ . Rezultat prezentuje się następująco:

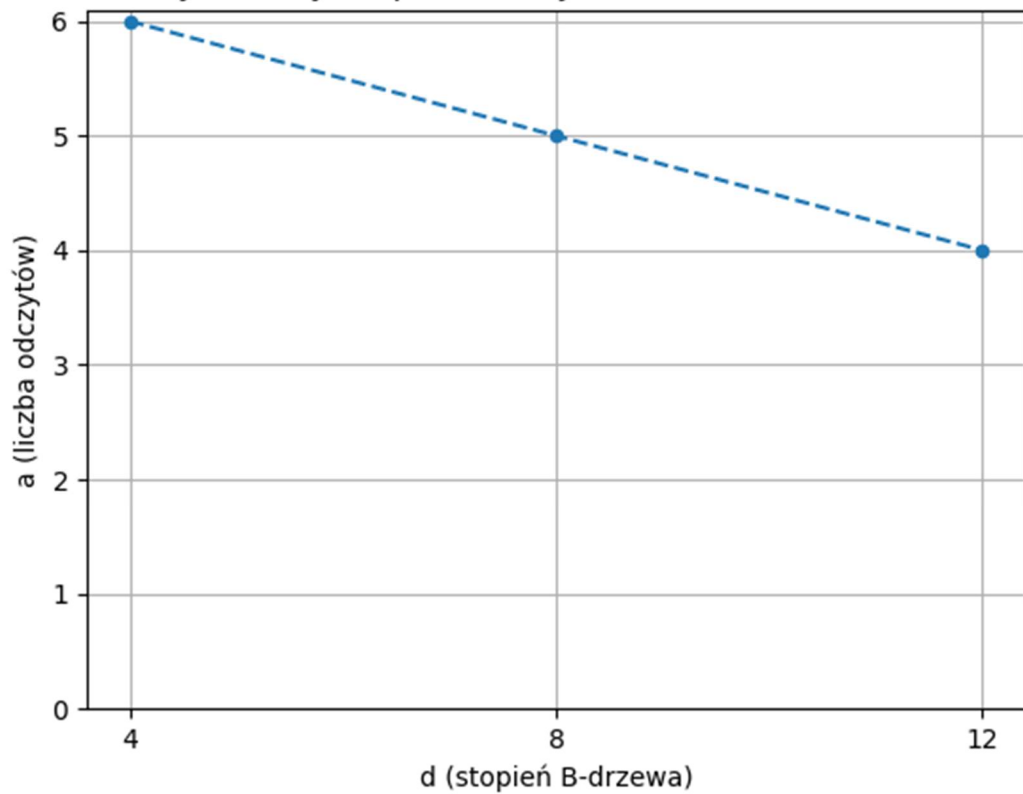
Liczba odczytów z dysku podczas wyszukiwania rekordu dla  $N = 100$



Liczba odczytów z dysku podczas wyszukiwania rekordu dla  $N = 1000$



Liczba odczytów z dysku podczas wyszukiwania rekordu dla  $N = 10000$



W tym przypadku widać tendencję nierosnącą liczby dostępów do dysku wraz ze wzrostem stopnia B-drzewa. Im większy stopień  $d$ , tym niższe jest drzewo, a zatem mniej stron należy pobrać do pamięci, by dojść do szukanego rekordu.