

# STRUKTURY DANYCH I ZŁOŻONOŚĆ OBLICZENIOWA

Zadanie projektowe nr 1: Badanie efektywności operacji na danych w podstawowych strukturach danych

*Uzupełnienie sprawozdania o testy operacji wyszukiwania*

Autor: Bartosz Rodziewicz, 226105

Prowadzący: dr inż. Dariusz Banasiak

Grupa: Wtorek, TN, 7:30

## 1. Wstęp teoretyczny – operacja wyszukiwania

### a. Tablica

Operacja wyszukiwania w tablicy polega na jednokrotnym przejściu po każdym elemencie. Złożoność obliczeniowa wynosi  $O(n)$ .

### b. Lista dwukierunkowa

Operacja wyszukiwania w tablicy polega na jednokrotnym przejściu po każdym elemencie. Złożoność obliczeniowa wynosi  $O(n)$ .

### c. Kopiec

Mimo dostępności lepszego algorytmu wyszukiwania danych w kopcu, zaimplementowałem zwykły algorytm przeszukiwania tablicy, ponieważ w trakcie realizacji projektu nie byłem w stanie znaleźć żadnych informacji o nim. Złożoność obliczeniowa wynosi  $O(n)$ . Dodatkowo, z uwagi na właściwość kopca dostęp do największej jego wartości nie wymaga wyszukiwania i posiada złożoność  $O(1)$ .

## 2. Metoda testowania i plan eksperymentu

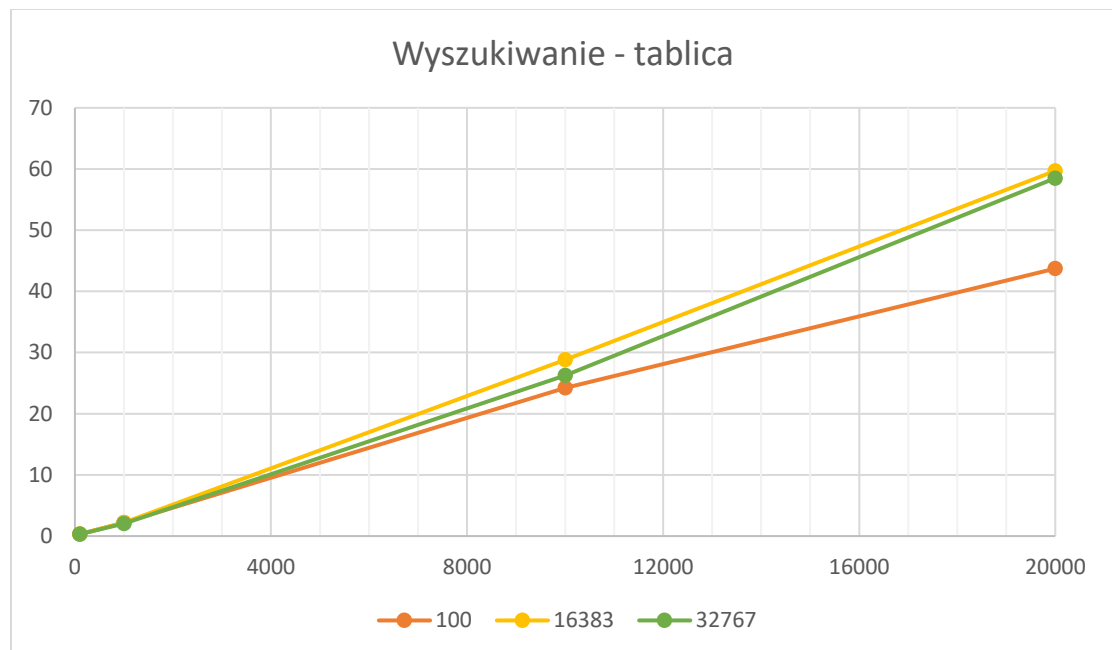
*Metoda testów jest identyczna jak w przypadku poprzednich.*

Test polega na stworzeniu struktury z konkretnego przypadku za pomocą metody generowania, wylosowaniu szukanej wartości, uruchomienie licznika, wykonanie wyszukiwania, zatrzymanie licznika, zapisanie wyniku i powtórzenie jej.

**Wyniki pomiarów podane są w mikrosekundach.**

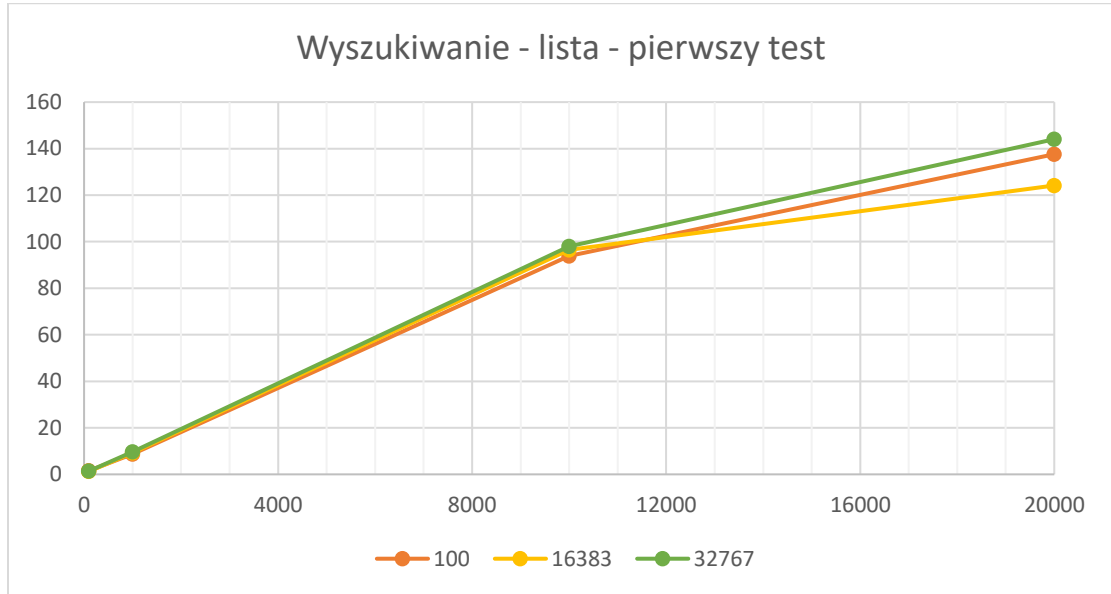
## 3. Wyniki testów – wyszukiwanie

### a. Tablica



Testy wyszukiwania w tablicy dają wyniki zgodne z teorią i złożoność obliczeniowa wychodzi  $O(n)$ . Lekkie wahnięcia następują z powodu większego obciążenia komputera w chwili pomiarów. Nie jestem jednak w stanie wyjaśnić faktu, dlaczego tablica wypełniona danymi z zakresu  $[0, 100]$ , gdzie o wiele częściej następuje znalezienie konkretnego elementu (i z tego powodu wykonanie dodatkowej operacji) jest najszybsze.

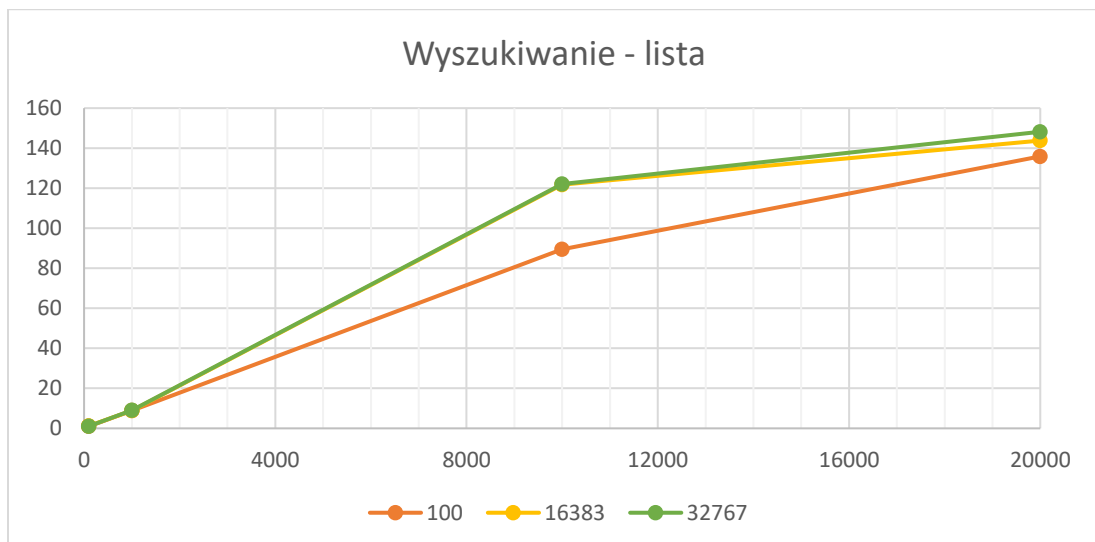
#### b. Lista dwukierunkowa



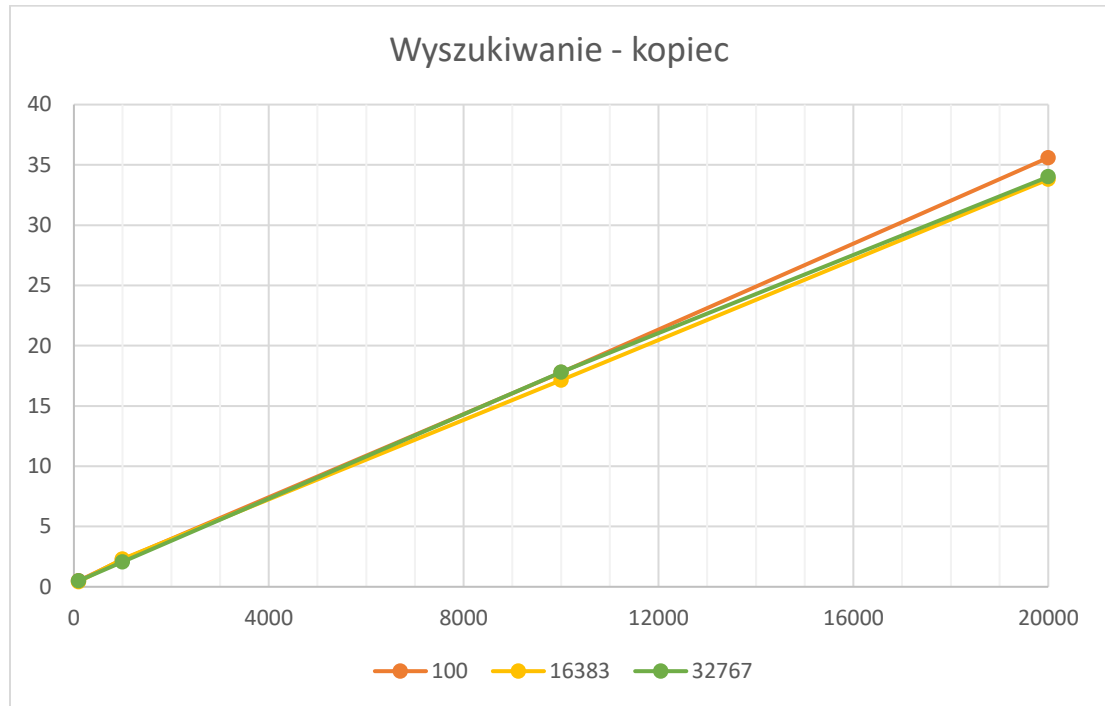
Test wyszukiwania elementu w liście przyniósł bardzo zaskakujące wyniki i bardziej przypomina mi  $O(\log(n))$ . Niestety nie jestem w stanie wytłumaczyć, dlaczego tak wyszło.

Dla pewności powtórzyłem badanie i wyniki wyszły jeszcze bardziej w stronę  $O(\log(n))$ .

Sytuacja staje się lekko bardziej klarowna, gdy porówna się te wyniki do innych struktur i widać, że lista zostaje bardzo w tyle co pozwala mi postawić tezę, że algorytm testowania wyszukiwania dla list mógł być błędny.

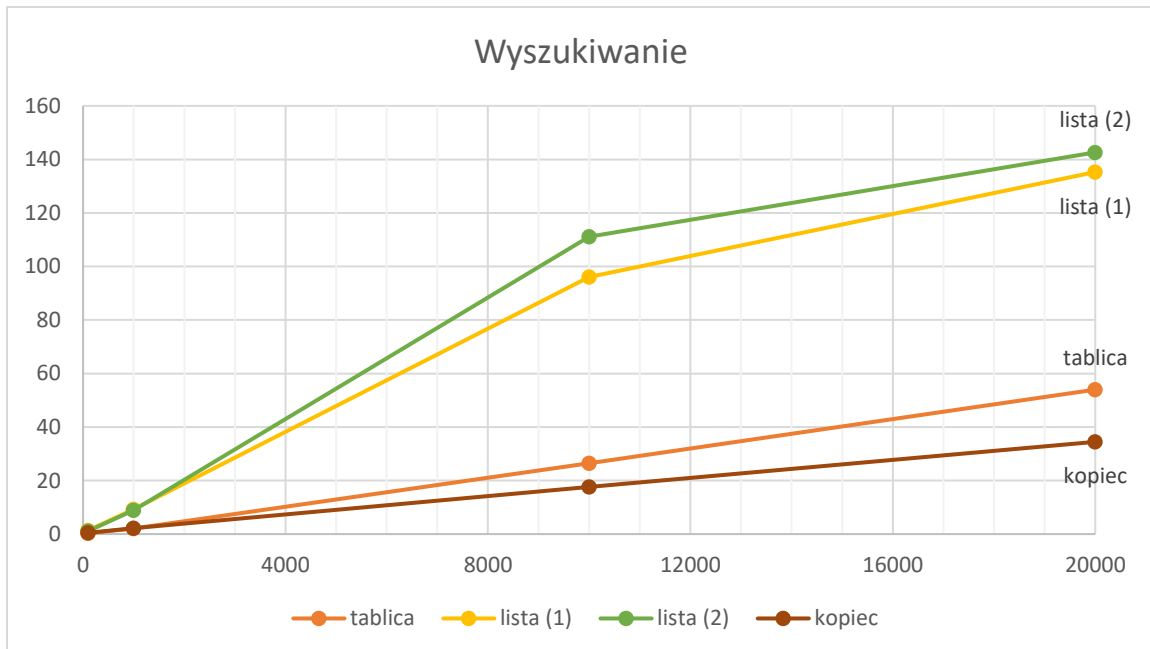


#### c. Kopiec



Tak jak napisałem we wstępie z uwagi na zastosowaną przeze mnie implementację wyszukiwania w formie tablicowej (dokładnie tak samo jak w przypadku tablicy) złożoność jej wychodzi  $O(n)$ , co powyższe testy potwierdzają.

#### 4. Porównanie struktur



Porównanie struktur pod kątem czasu wyszukiwania pokazuje bardzo dziwne wyniki, które ciężko mi jakkolwiek przeanalizować. Tak jak pisałem wcześniej wyniki listy wyszły bardzo dziwnie. Nie rozumiem również dlaczego wyniki tablicy i kopca są aż tak rozbieżne – to dokładnie taki sam algorytm przeszukujący dokładnie taką samą tablicę o takim samym rozkładzie i wielkości.

## 5. Wnioski

- Ciężko wypisać mi jakieś wnioski z tego patrząc na to jak wyniki się ułożyły.
- Nie jestem w stanie zdiagnozować, dlaczego jest tak duża rozbieżność pomiędzy wynikami tablicy i kopca oraz dlaczego wyniki listy „pokazały” inną złożoność.
- Analizując pojedyncze wyniki z każdego pomiaru osobno (załączone do sprawozdania na końcu) zauważyłem, że test wyszukiwania dawał bardzo dużą rozbieżność pomiędzy najmniejszą, a największą wartością, podczas gdy poprzednie testy zachowywały w miarę spójne wyniki.