

Sprawozdanie

Złożone struktury danych:

listy i drzewo BST

Używane języki: C/C++

Xenia Pietrzyk
nr indeksu: 146409
Kamil Lewiński
nr indeksu: 146437
grupa 4B

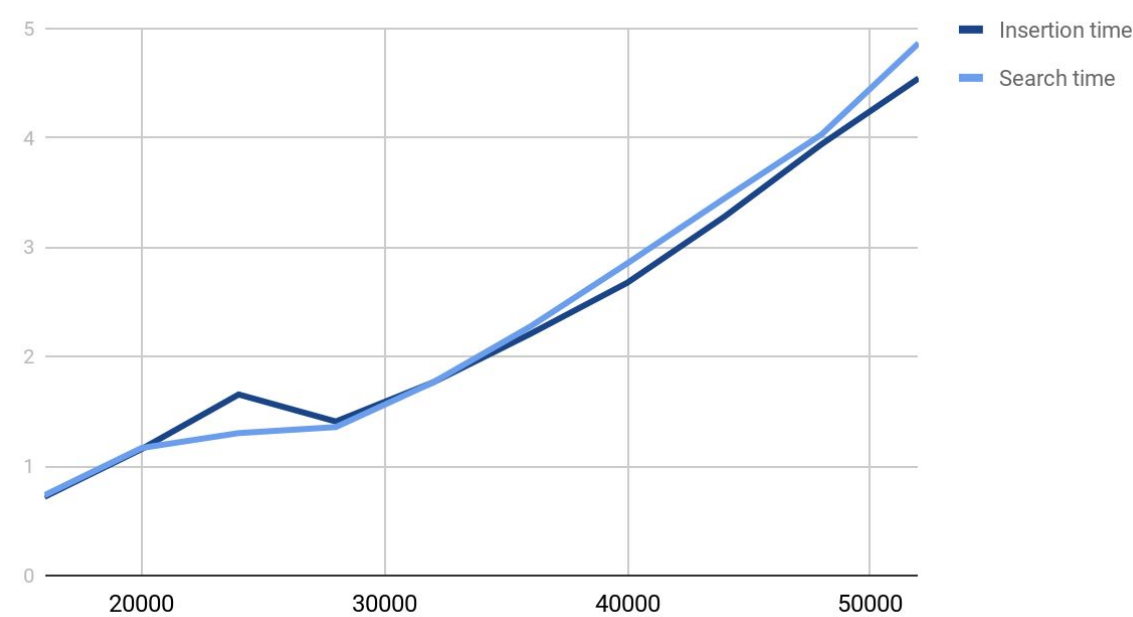
Listy

Increase

	16k	20k	24k	28k	32k	36k	40k	44k	48k	52k
Insertion time	0.719	1.156	1.655	1.408	1.765	2.208	2.676	3.280	3.942	4.545
Search time	0.734	1.166	1.302	1.357	1.765	2.275	2.853	3.447	4.031	4.866

Tab.1. Tabela obrazująca czas realizacji dodawania oraz wyszukiwania elementów w liście uporządkowanej rosnąco.

Points scored

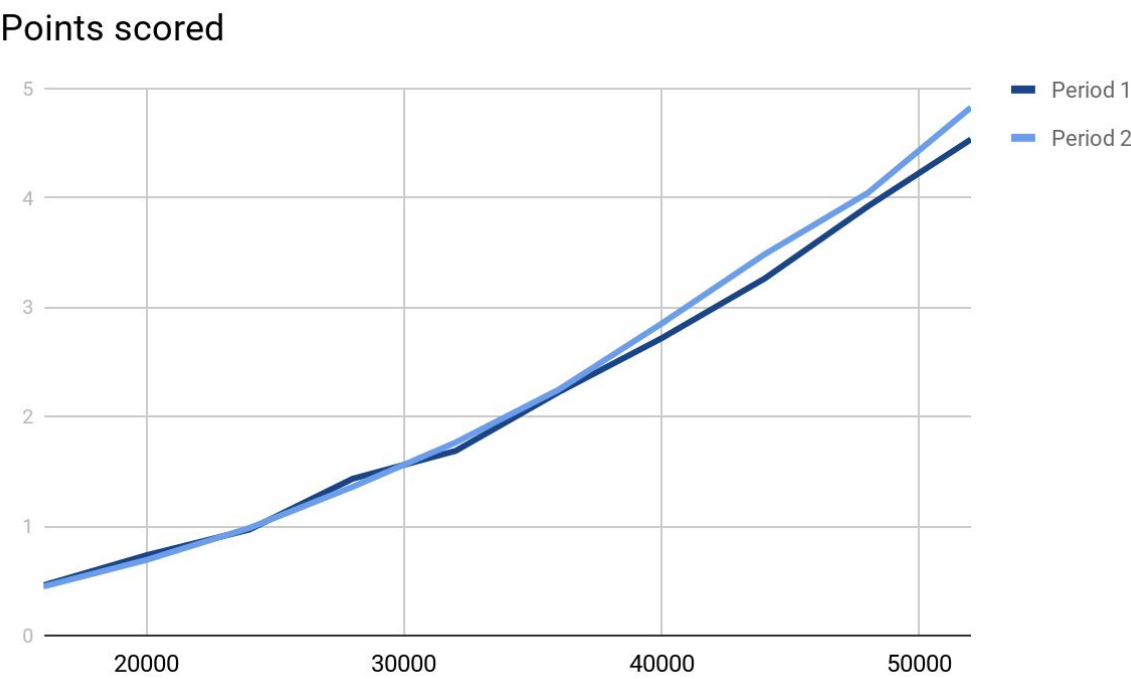


Wykres obrazujący dane z powyższej tabeli Tab.1.

Random

	16k	20k	24k	28k	32k	36k	40k	44k	48k	52k
Insertion time	0,459	0,734	0,969	1,433	1,688	2,225	2,718	3,264	3,923	4,535
Search time	0,448	0,691	0,984	1,359	1,765	2,248	2,852	3,486	4,046	4,827

Tab.2. Tabela obrazująca czas realizacji dodawania oraz wyszukiwania elementów w liście uporządkowanej losowo.



Wykres obrazujący dane z powyższej tabeli Tab.2.

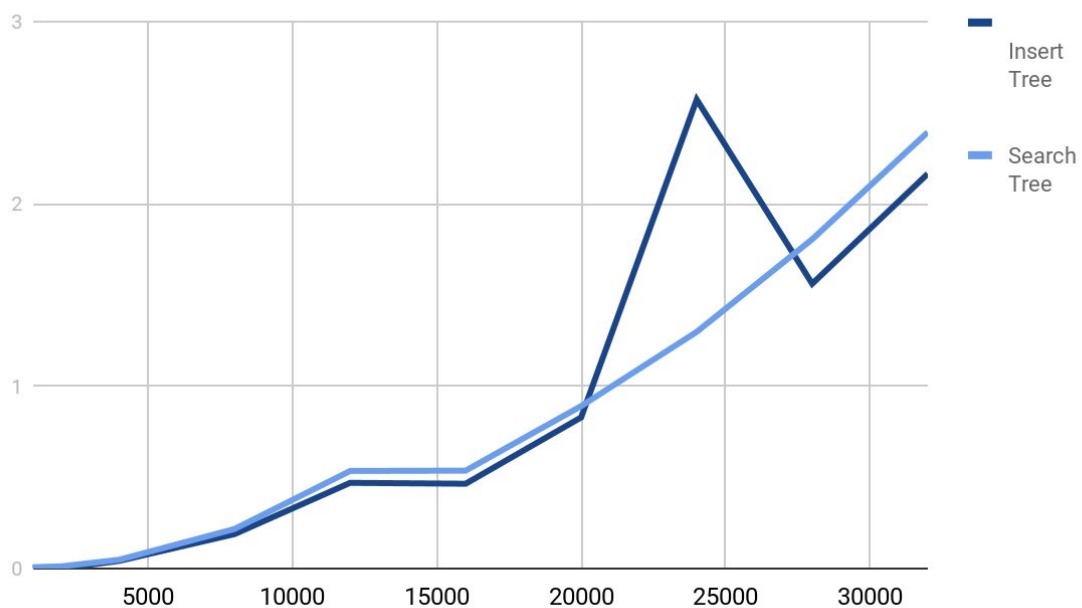
Drzewo BST

Increasing

	1k	2k	4k	8k	12k	16k	20k	24k	28k	32k
Insert Tree	0,002	0,013	0,042	0,189	0,470	0,465	0,829	2,572	1,562	2,165
Search Tree	0,007	0,012	0,048	0,217	0,535	0,537	0,891	1,298	1,806	2,393

Tab.3. Tabela obrazująca czas realizacji dodawania oraz wyszukiwania elementów w drzewie uporządkowanym rosnąco.

Points scored



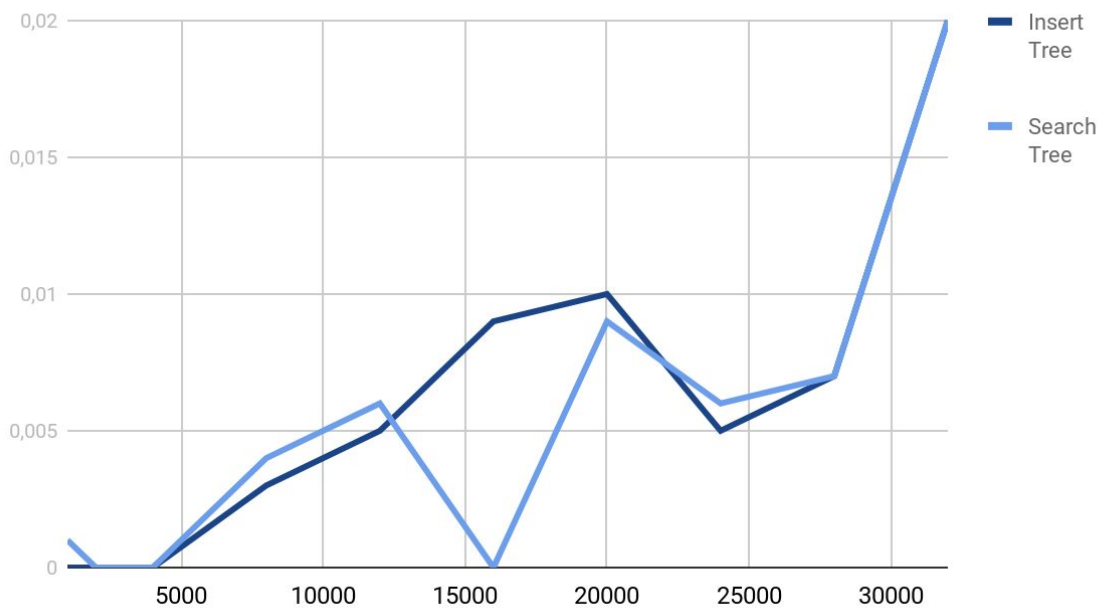
Wykres obrazujący dane z powyższej tabeli Tab.3.

Random

	1k	2k	4k	8k	12k	16k	20k	24k	28k	32k
Insert Tree	0,000	0,000	0,000	0,003	0,005	0,009	0,010	0,005	0,007	0,020
Search Tree	0,001	0,000	0,000	0,004	0,006	0,000	0,009	0,006	0,007	0,020

Tab.4. Tabela obrazująca czas realizacji dodawania oraz wyszukiwania elementów w drzewie uporządkowanym losowo..

Points scored



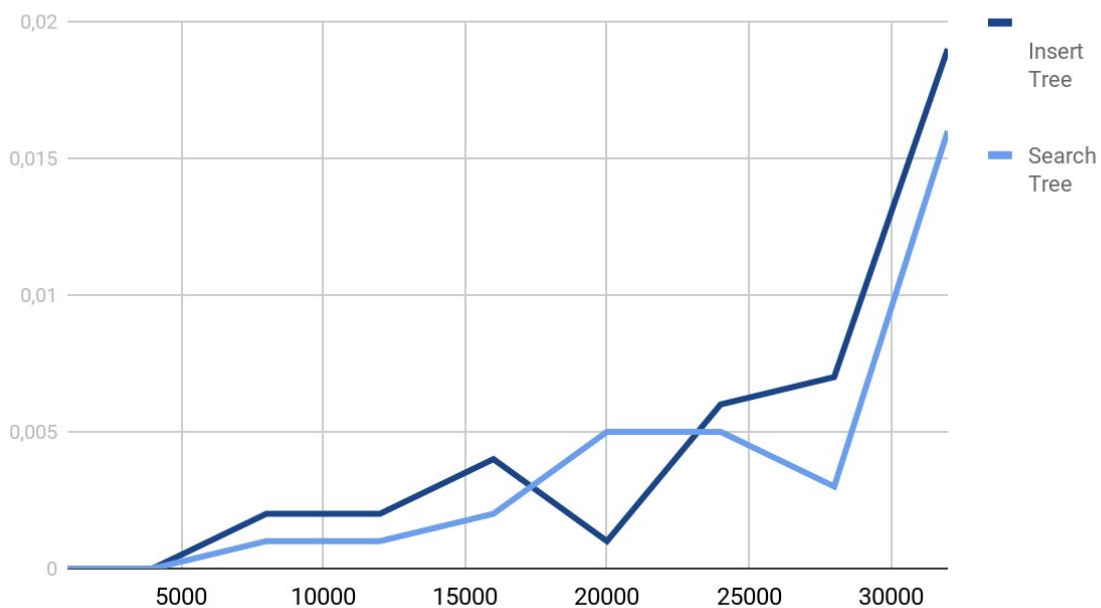
Wykres obrazujący dane z powyższej tabeli Tab.4.

Binary

	1k	2k	4k	8k	12k	16k	20k	24k	28k	32k
Insert Tree	0,000	0,000	0,000	0,002	0,002	0,004	0,001	0,006	0,007	0,019
Search Tree	0,000	0,000	0,000	0,001	0,001	0,002	0,005	0,005	0,003	0,016

Tab.5. Tabela obrazująca czas realizacji dodawania oraz wyszukiwania elementów w drzewie uporządkowanym binarnie.

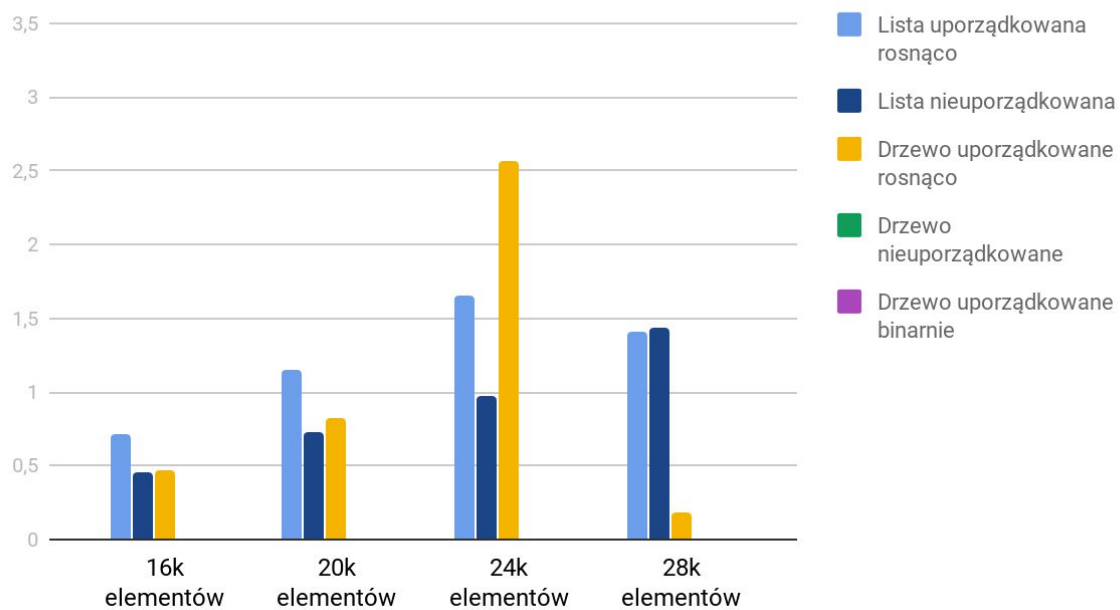
Points scored



Wykres obrazujący dane z powyższej tabeli Tab.5.

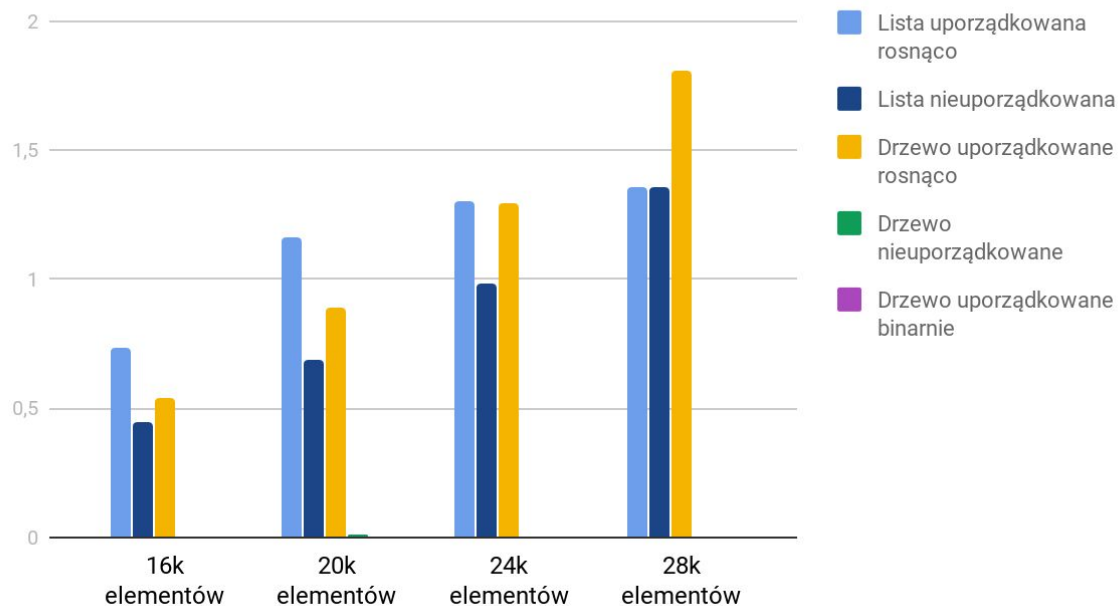
Porównanie

Points scored



Wykres porównujący uzyskane wyniki dla danych wizualizujących "Insert Tree".

Points scored



Wykres porównujący uzyskane wyniki dla danych wizualizujących "Search Tree".

Wnioski: Struktura Lista w większości przeanalizowanych przypadków okazała się znacząco mniej korzystna od struktury Drzewo BST. Uzyskane wyniki dla list są około stukrotnie wyższe od wyników dla drzew. Wyjątkiem są wyniki uzyskane dla drzew uporządkowanych rosnąco. Widoczna jest również rozbieżność na korzyść listy uporządkowanej rosnąco w porównaniu do listy nieuporządkowanej. Największą pomiędzy nimi rozbieżność zaobserwowano w przypadku analizy tablicy o 24-tysiącach elementów.

Uzasadnieniem naszych wyników jest różnica w strukturze uporządkowania danych. Dla listy aby przejść do danego elementu musimy przejść przez każdy element znajdujący się przed szukanym elementem, a w Drzewie BST dane są dzielone co minimalizuje liczbę operacji potrzebnych do znalezienia szukanego elementu. Wyjątkiem jest jednak przypadek gdy porządkujemy dane rosnąco. Wtedy struktura drzewa tak naprawdę niewiele różni się od struktury listy ponieważ wszystkie kolejne elementy znajdują się w jednej odnodze drzewa.

Błędy pomiarowe mogą wynikać z procesów niezależnych działających w tle w systemie. Zaobserwowaliśmy różnicę pomiędzy wynikami podczas włączania różnych aplikacji.