

SPRAWOZDANIE

Złożone struktury danych Listy jednokierunkowe i drzewa BST

Maciej Handke 146549

Michał Janas 146363

Pomiary czasów zostały wykonane na sprzęcie w poniższej konfiguracji:

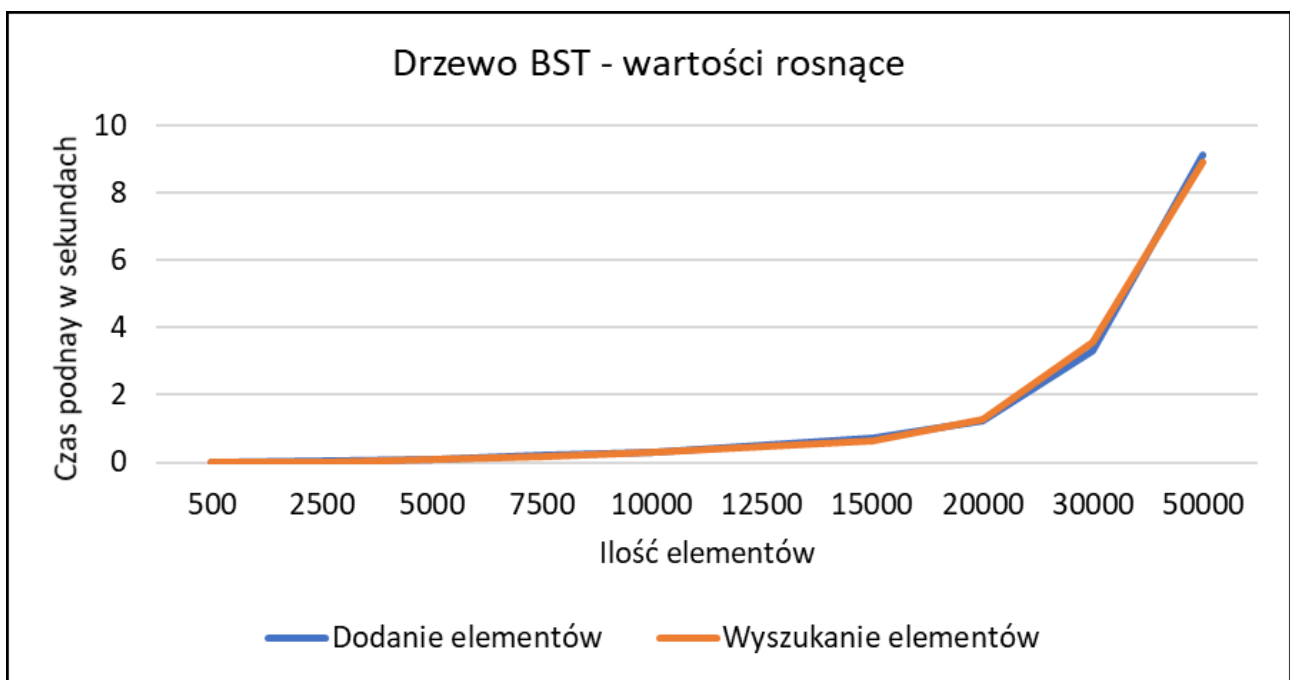
System: Windows 10 x64

Procesor: AMD Ryzen™ 5 2500U

Pamięć RAM: 8,0 GB

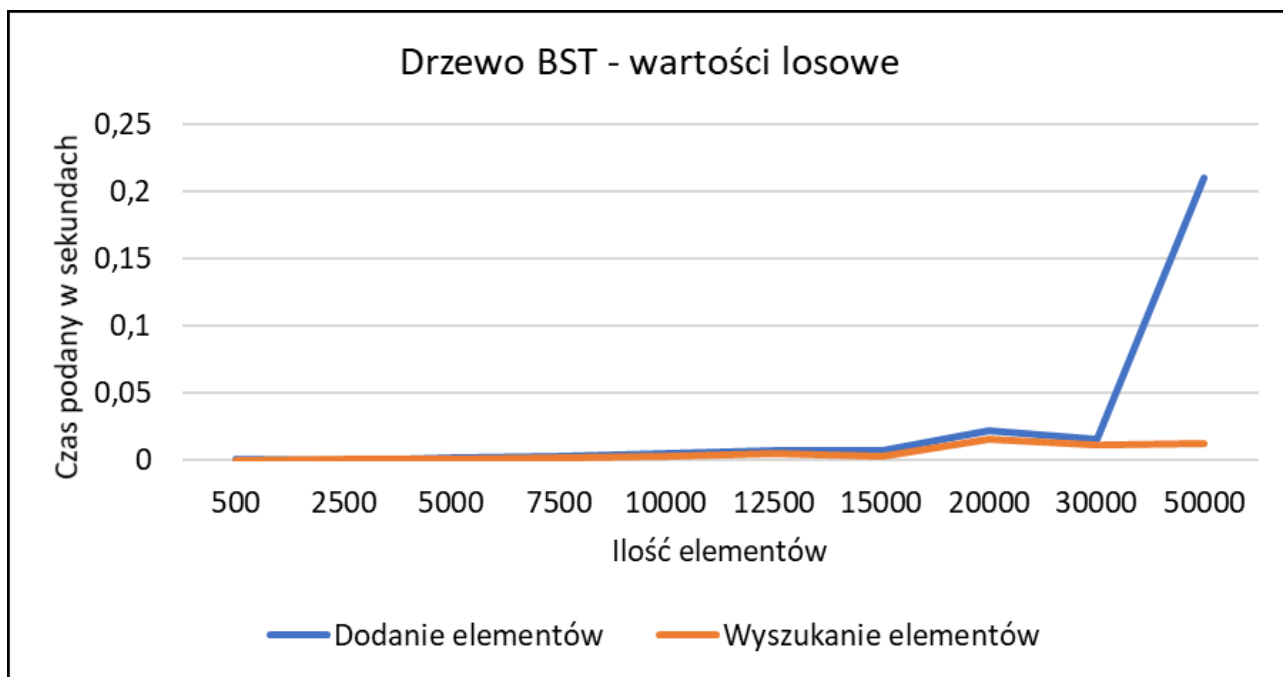
Poniższe tabele i wykresy zawierają pomiary czasów dla dodania i wyszukania elementów w listach jednokierunkowych i drzewach BST dla instancji o zdefiniowanych wielkościach

Drzewo BST - wartości rosnące		
Ilość elementów	Dodanie elementów	Wyszukanie elementów
500	0,001	0,002
2500	0,022	0,019
5000	0,078	0,074
7500	0,195	0,167
10000	0,316	0,296
12500	0,514	0,478
15000	0,705	0,651
20000	1,249	1,263
30000	3,303	3,563
50000	9,124	8,93
Czas podany w sekundach		



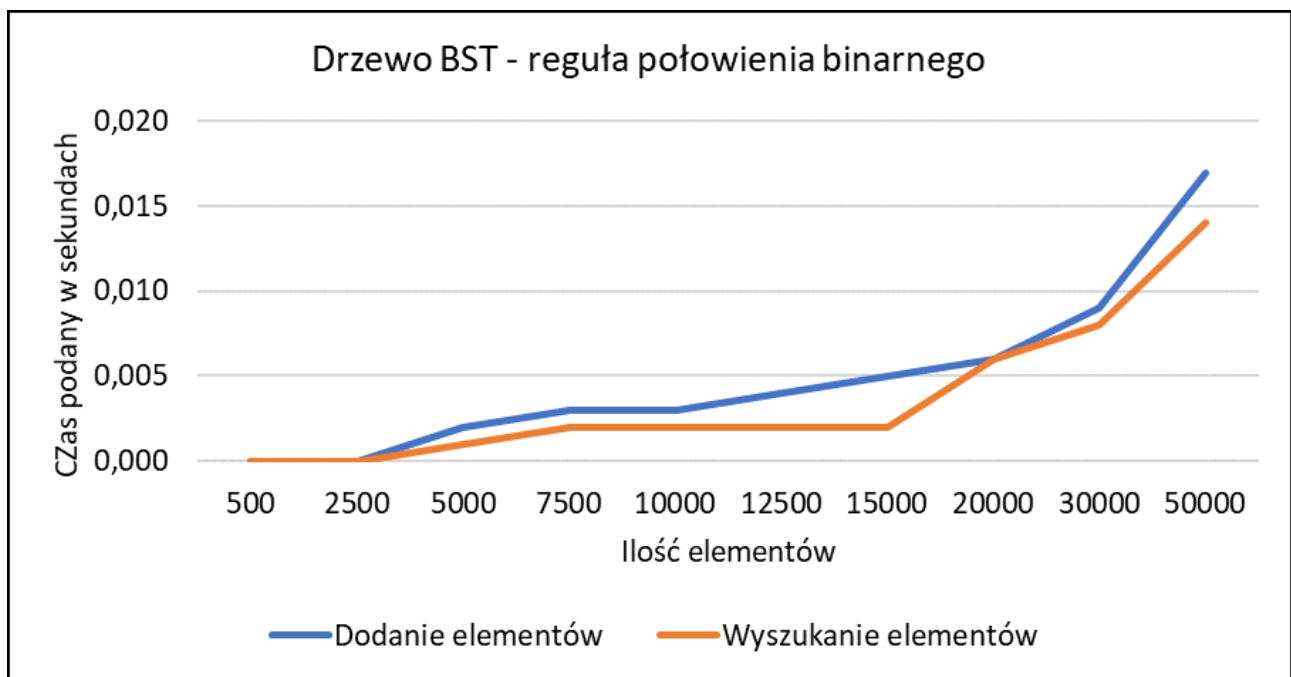
Czas dodawania jak i wyszukiwania elementów jest podobny. Ponadto rośnie on dla obu przypadków mniej więcej w sposób jednostajny

Drzewo BST - wartości losowe		
Ilość elementów	Dodanie elementów	Wyszukanie elementów
500	0,001	0,000
2500	0,000	0,001
5000	0,002	0,001
7500	0,003	0,002
10000	0,005	0,003
12500	0,007	0,005
15000	0,007	0,003
20000	0,022	0,016
30000	0,015	0,011
50000	0,210	0,012
Czas podany w sekundach		



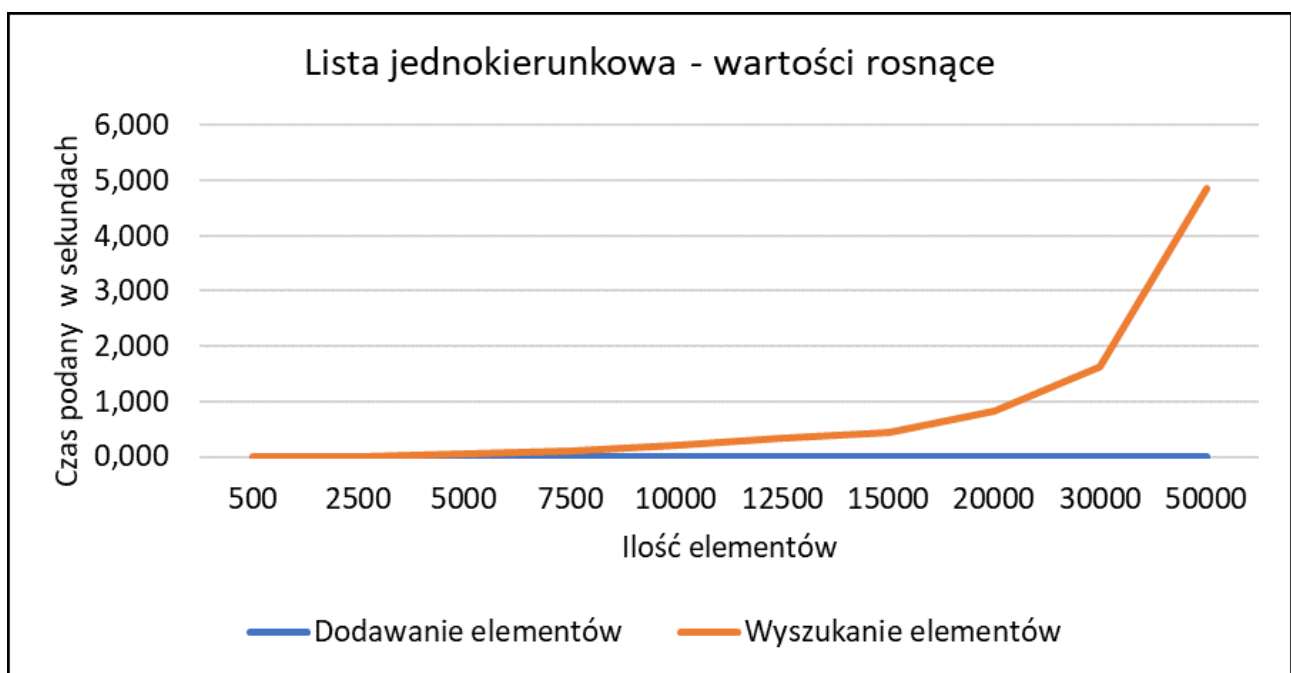
Dla wartości losowych czas dodania elementów jest także podobny. Różnicą jest instancja z 50000 elementów. W tym przypadku dodanie trwa znacznie dłużej niż wyszukiwanie

Drzewo BST - reguła połowienia binarnego		
Ilość elementów	Dodanie elementów	Wyszukanie elementów
500	0,000	0,000
2500	0,000	0,000
5000	0,002	0,001
7500	0,003	0,002
10000	0,003	0,002
12500	0,004	0,002
15000	0,005	0,002
20000	0,006	0,006
30000	0,009	0,008
50000	0,017	0,014
Czas podany w sekundach		



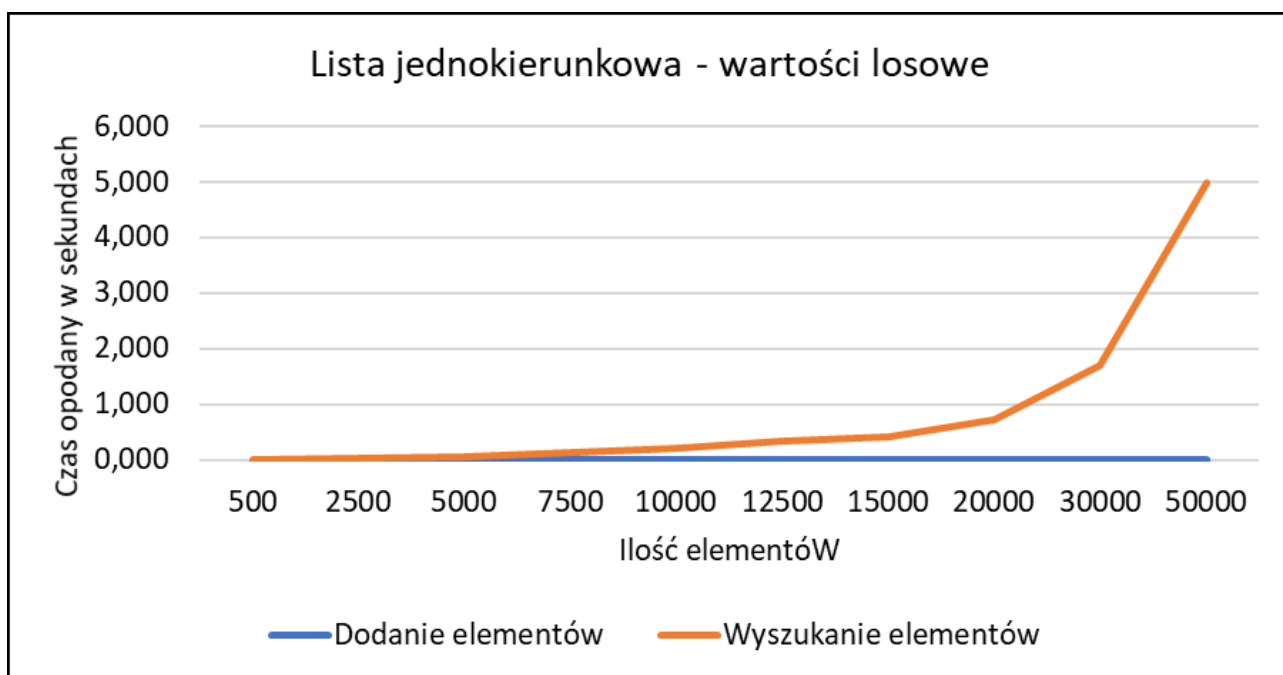
Posiłkując się wykresem, w przypadku reguły połowienia binarnego, czas wyszukania elementów jest zauważalnie niższy niż ma to miejsce przy ich dodawaniu

Lista jednokierunkowa - wartości rosnące		
Ilość elementów	Dodanie elementów	Wyszukanie elementów
500	0,000	0,000
2500	0,001	0,011
5000	0,000	0,054
7500	0,001	0,112
10000	0,001	0,205
12500	0,002	0,328
15000	0,001	0,437
20000	0,001	0,829
30000	0,003	1,628
50000	0,005	4,850
Czas podany w sekundach		



W przeciwieństwie do drzew BST, różnica w wyszukiwaniu i dodaniu elementów jest znaczna. Dodanie to zaledwie ułamek czasu, którego potrzeba na jego znalezienie. Widać to szczególnie dla instancji o dużej ilości elementów

Lista jednokierunkowa - wartości losowe		
Ilość elementów	Dodanie elementów	Wyszukanie elementów
500	0,000	0,001
2500	0,000	0,018
5000	0,001	0,063
7500	0,001	0,118
10000	0,001	0,210
12500	0,001	0,325
15000	0,001	0,407
20000	0,002	0,725
30000	0,004	1,690
50000	0,007	4,973
Czas podany w sekundach		



Podobnie jak w listach jednokierunkowych z wartościami rosnącymi, w wartościach losowych różnica pomiędzy wyszukianiem a dodaniem jest znacząca

Wnioski

W przypadku drzew BST, największą wydajnością charakteryzują się ułożenia danych w sposób losowy oraz binarny. Różnica pomiędzy nimi a wartościami rosnącymi, które są zdecydowanie najwolniejsze jest pokaźna.

W odróżnieniu do drzew BST, różnica w wyszukiwaniu i dodaniu elementów jest o wiele bardziej widoczna. Czas dodania elementu to mała część tego, co potrzeba na jego znalezienie. Widoczne jest to szczególnie dla instancji zawierającej dużą ilość elementów.

Porównując ułożenie danych w sposób losowy, wydajność w wyszukiwaniu elementów listy jednokierunkowej jest kolosalnie gorsza niż w przypadku drzewa BST, natomiast odwrotnie jest w dodawaniu elementów.

Sprawa ma się inaczej w przypadku wartości rosnących. Dla wyszukiwania elementów list jednokierunkowych wydajność w czasie działania jest dwukrotnie lepsza, porównując je do drzew BST, a różnica w dodawaniu elementów jest ogromna, na korzyść dla list jednokierunkowych.