Sprawozdanie

Autor: Sławomir Staniszewski

Nr albumu: 135893

Grupa dziekańska: 3a

Pomiary wykonano na:
AMD Ryzen 7 2700X, 3.7 GHz, 8/16; 16 GB RAM

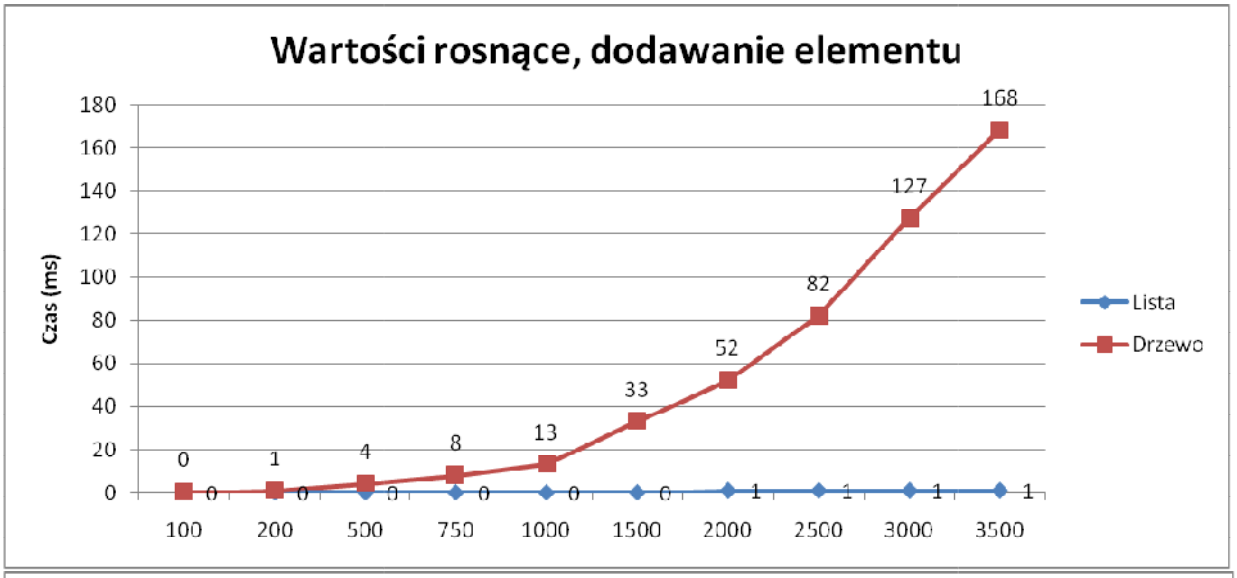
Algorytmy i Struktury Danych

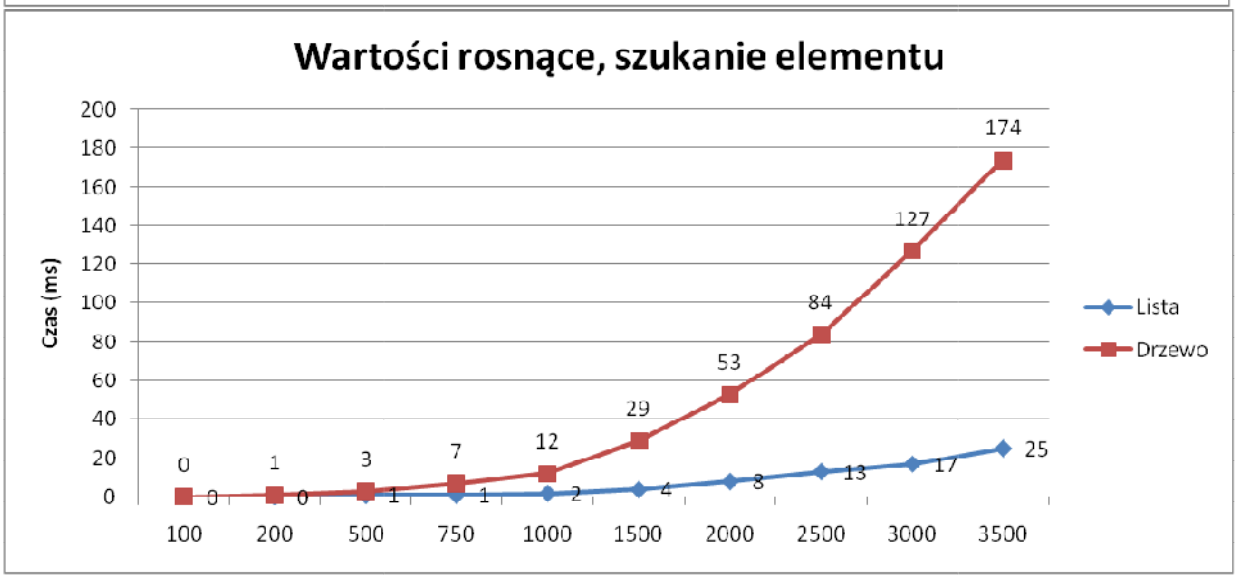
Temat: Złożone struktury danych

Pomiar czasu dodania / wyszukania elementów dla wartości rosnących:

W. rosnące	Lista		Drzewo	
Próbka	Dodawanie	Szukanie	Dodawanie	Szukanie
100	0	0	0	0
200	0	0	1	1
500	0	1	4	3
750	0	1	8	7
1000	0	2	13	12
1500	0	4	33	29
2000	1	8	52	53
2500	1	13	82	84
3000	1	17	127	127
3500	1	25	168	174

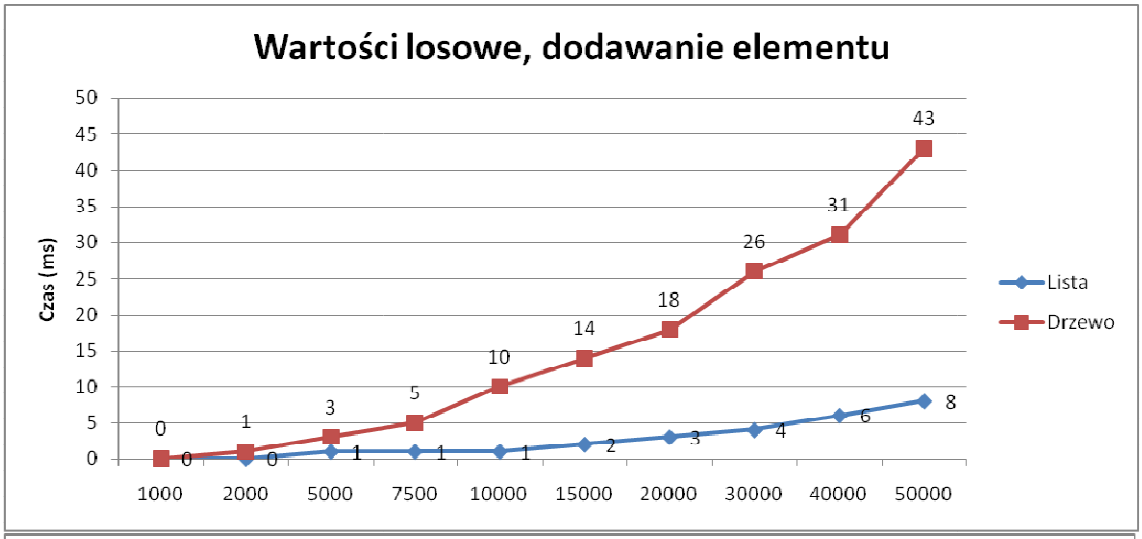
W. rosnące	Lista	
Próbka	Dodawanie	Szukanie
1000	0	2
2000	0	8
5000	1	47
7500	1	109
10000	2	215
15000	2	543
20000	3	1 033
30000	4	2 341
40000	6	4 220
50000	8	6 788

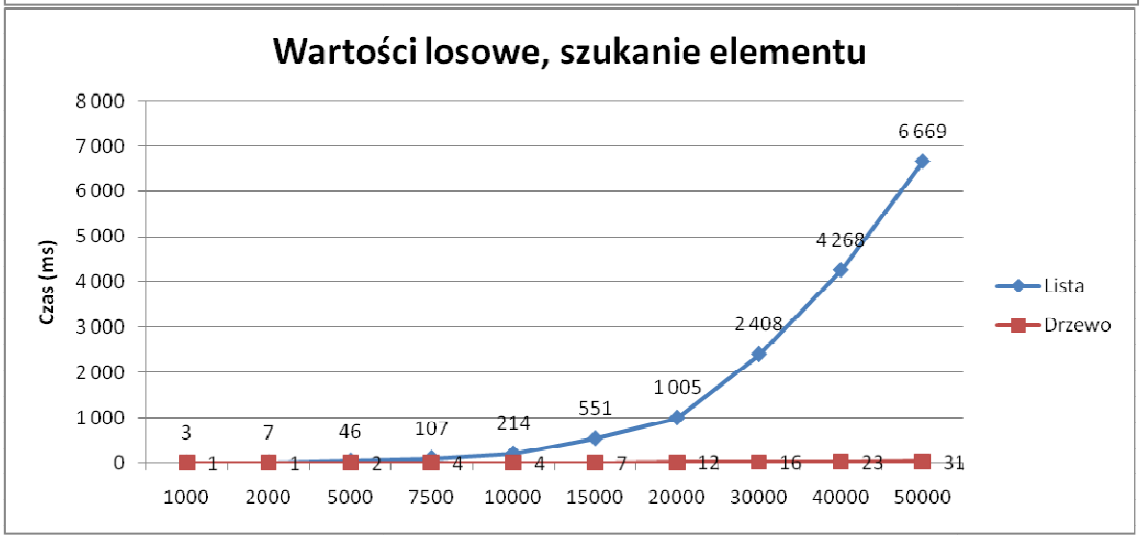




Pomiar czasu dodania / wyszukania elementów dla wartości losowych:

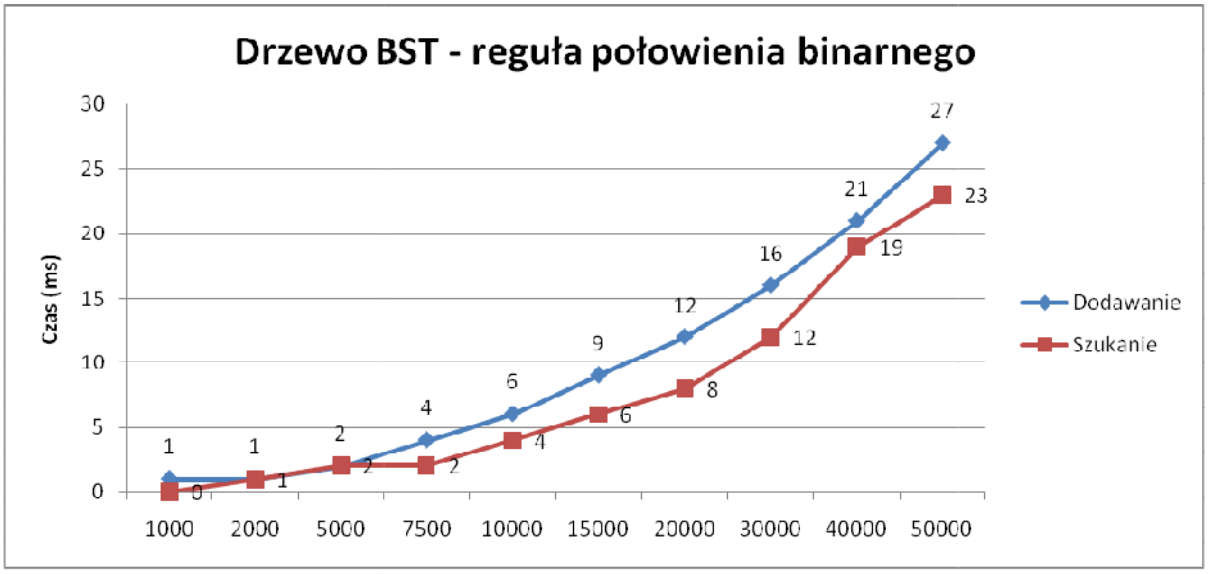
W. losowe	Lista		Drzewo	
Próbka	Dodawanie	Szukanie	Dodawanie	Szukanie
1000	0	3	0	1
2000	0	7	1	1
5000	1	46	3	2
7500	1	107	5	4
10000	1	214	10	4
15000	2	551	14	7
20000	3	1 005	18	12
30000	4	2 408	26	16
40000	6	4 268	31	23
50000	8	6 669	43	31





Pomiar czasu dodania / wyszukania elementów dla wartości binarnych:

W. binarne	Drzewo	
Próbka	Dodawanie	Szukanie
1000	1	0
2000	1	1
5000	2	2
7500	4	2
10000	6	4
15000	9	6
20000	12	8
30000	16	12
40000	21	19
50000	27	23



Podsumowanie

Wartości rosnące – Zarówno dodawanie i wyszukanie wartości rosnących było zdecydowanie szybsze w liście niż w drzewie BST.
Niemożliwe okazało się przeprowadzenie próby dla drzewa BST przy wielkości elementów przekraczającej 4000. Dochodziło do przepełnienia stosu.

Wartości losowe – Dodawanie elementów do listy okazało się kilkukrotnie szybsze niż dodawanie do drzewa. Nadal jednak pozostajemy na poziomie milisekund, nawet dla 50000 elementów.
Operacje polegające na wyszukiwaniu były znacznie szybsze w drzewie niż na liście. Już przy 10000 elementów czas dla listy wyniósł pełną sekundę, podczas gdy w drzewie wciąż było to tylko kilkanaście milisekund.

Wartości dodawane zgodnie z regułą połowienia binarnego – Wraz ze wzrostem próbki pogłębia się różnica między dodaniem i wyszukaniem elementu, na korzyść tego pierwszego. Operowanie na wartościach dobranych w ten sposób okazało się najbardziej optymalne. Widać znaczącą poprawę w stosunku do wartości losowych i kilkukrotną poprawę przy wartościach rosnących.

Najbardziej optymalne okazuje się korzystanie z drzewa BST. Jest bardziej stabilne, w żadnym przypadku operacje nie przekroczyły 1 sekundy. Lista jest lepsza tylko dodawaniu wartości – podczas ich wyszukiwania występują kolosalne straty w czasie wraz ze wzrostem próbki.