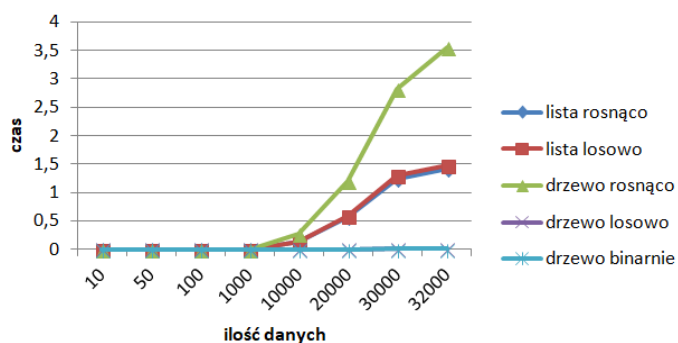


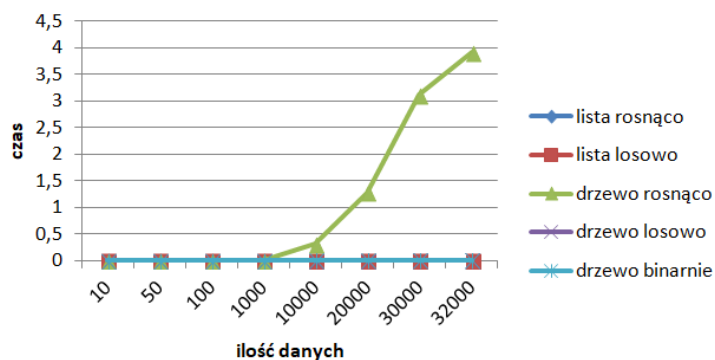
struktura	ułożenie danych	ilość danych	czas wstawiania	czas wyszukiwania
lista	rosnąco	10	0	0
		50	0	0
		100	0	0
		1000	0	0,002
		10000	0	0,135
		20000	0,001	0,566
		30000	0,001	1,236
		32000	0,001	1,409
	losowo	10	0	0
		50	0	0
		100	0	0
		1000	0	0,002
		10000	0,001	0,141
		20000	0,001	0,579
		30000	0,001	1,298
		32000	0,001	1,477
drzewo	rosnąco	10	0	0
		50	0	0
		100	0	0
		1000	0,003	0,002
		10000	0,317	0,268
		20000	1,279	1,199
		30000	3,107	2,827
		32000	3,917	3,553
	losowo	10	0	0
		50	0	0
		100	0	0
		1000	0	0
		10000	0,003	0,001
		20000	0,005	0,004
		30000	0,01	0,007
		32000	0,008	0,007
	binarnie	10	0	0
		50	0	0
		100	0	0
		1000	0	0,001
		10000	0,002	0,001
		20000	0,002	0,003
		30000	0,004	0,005
		32000	0,005	0,005

Zebrane dane do wykresów poniżej.

czas wyszukiwania

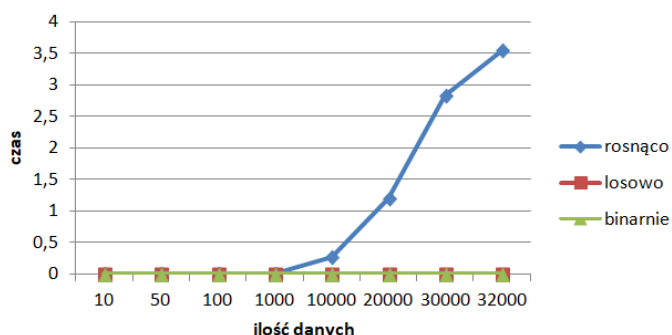


czasy wstawiania

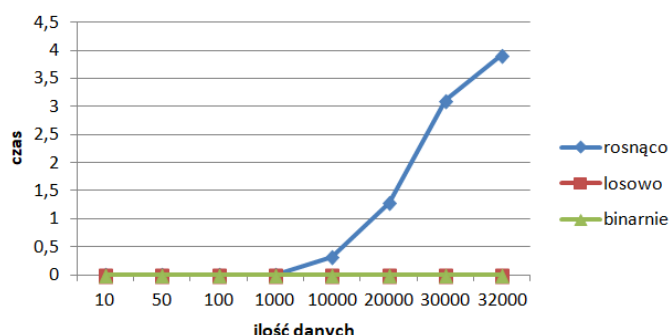


Wykresy porównawcze czasów wyszukiwania i wstawiania dla różnych układów danych na strukturach.

drzewo czas wyszukiwania

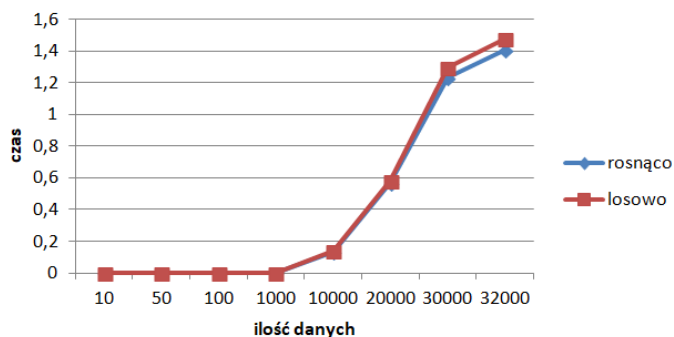


drzewo czas wstawiania

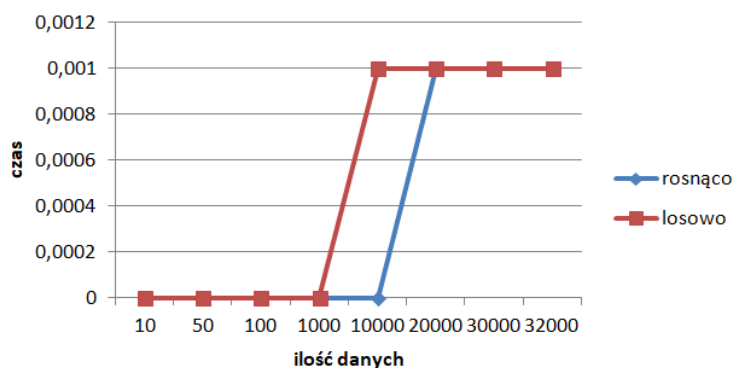


Wykresy porównawcze czasów wyszukiwania i wstawiania dla różnych układów danych dla drzew.

lista czas wyszukiwania



lista czas wstawiania



Wykresy porównawcze czasów wyszukiwania i wstawiania dla różnych układów danych w listach.

Wnioski:

W przypadku list czas działania algorytmów jest mniej więcej porównywalny dla obu rozkładów danych, zaś przy drzewach wyraźnie najwolniejsza opcja w obu przypadkach pomiarów jest rozkład danych rosnących. Porównując struktury, przy czasach wstawiania nie ma dużej różnicy dla rozkładu danych losowych, również rozłożenie binarne w drzewie ma pomiar bliski 0, natomiast przy rozkładowi rosnącemu lista wypada dużo lepiej od drzewa. Sprawa ma się nieco odwrotnie przy czasach wyszukiwania, gdzie w prawdzie najwolniej wypada ponownie drzewo o rozkładzie rosnącym, jednak pozostałe dwa układy danych przy drzewie są znacząco szybsze od wszystkich układów przy listach.