

Algorytmy i struktury danych

Sprawozdanie 2 – lista i drzewo.

Zuzanna Spychała

nr indeksu 145151

Olga Ormańczyk

nr indeksu 145183

Drzewo binarne

Tabela z czasami działania algorytmu dla elementów ułożonych rosnąco

[liczba elementów]	Dodawanie elementów	Wyszukiwanie elementów
1000	0.002000	0.003000
2000	0.011000	0.010000
3000	0.024000	0.025000
4000	0.044000	0.042000
5000	0.069000	0.067000
6000	0.099000	0.099000
7000	0.133000	0.134000
8000	0.177000	0.180000
9000	0.224000	0.226000
10000	0.289000	0.280000
11000	0.342000	0.340000
12000	0.405000	0.404000
13000	0.480000	0.475000
14000	0.553000	0.549000
15000	0.637000	0.636000
16000	0.733000	0.727000
17000	0.823000	0.854000
18000	0.913000	0.912000
19000	1.061000	1.040000
20000	1.137000	1.136000
21000	1.255000	1.249000
22000	1.364000	1.364000
23000	1.503000	1.488000

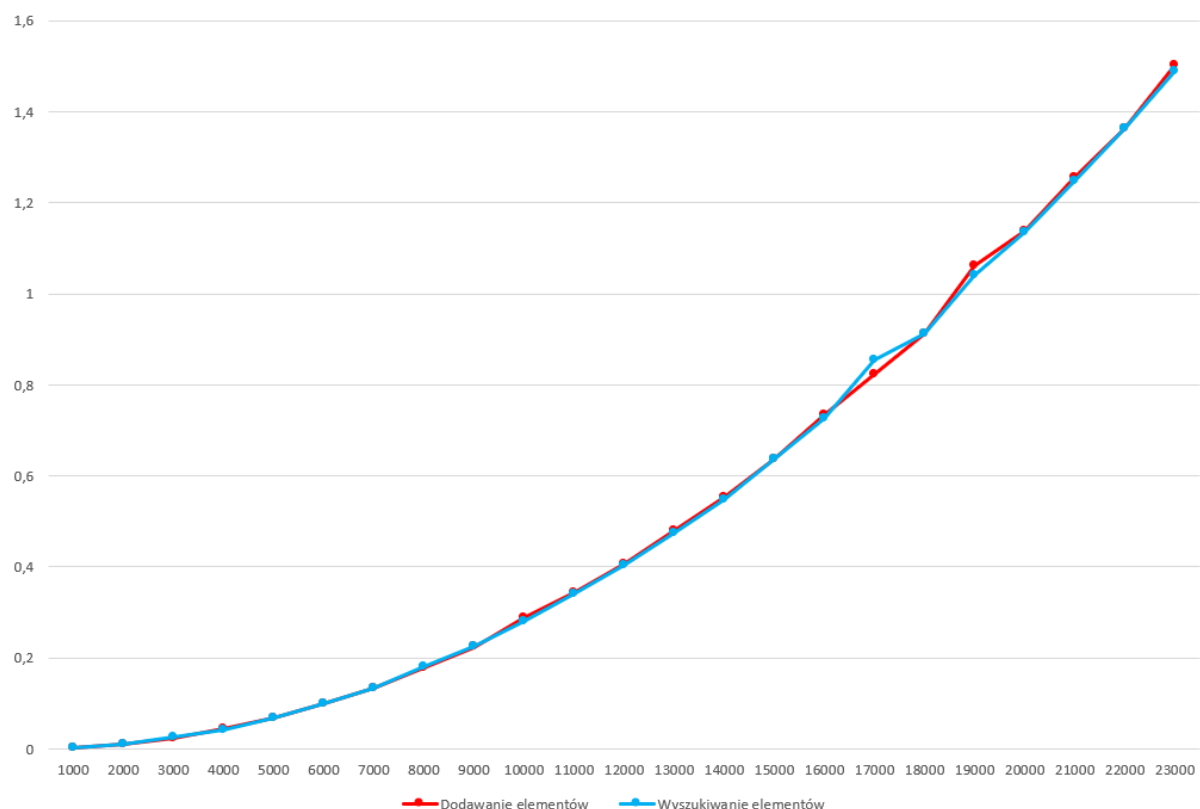


Tabela z czasami działania algorytmu dla elementów ułożonych losowo

[liczba elementów]	Dodawanie elementów	Wyszukiwanie elementów
1000	0.000000	0.000000
2000	0.001000	0.000000
3000	0.001000	0.000000
4000	0.001000	0.001000
5000	0.001000	0.001000
6000	0.001000	0.001000
7000	0.001000	0.001000
8000	0.002000	0.001000
9000	0.002000	0.002000
10000	0.002000	0.002000
11000	0.002000	0.001000
12000	0.003000	0.002000
13000	0.003000	0.003000
14000	0.003000	0.002000
15000	0.004000	0.003000
16000	0.004000	0.002000
17000	0.004000	0.003000
18000	0.004000	0.003000
19000	0.003000	0.004000
20000	0.005000	0.003000
21000	0.005000	0.004000
22000	0.005000	0.004000
23000	0.005000	0.004000

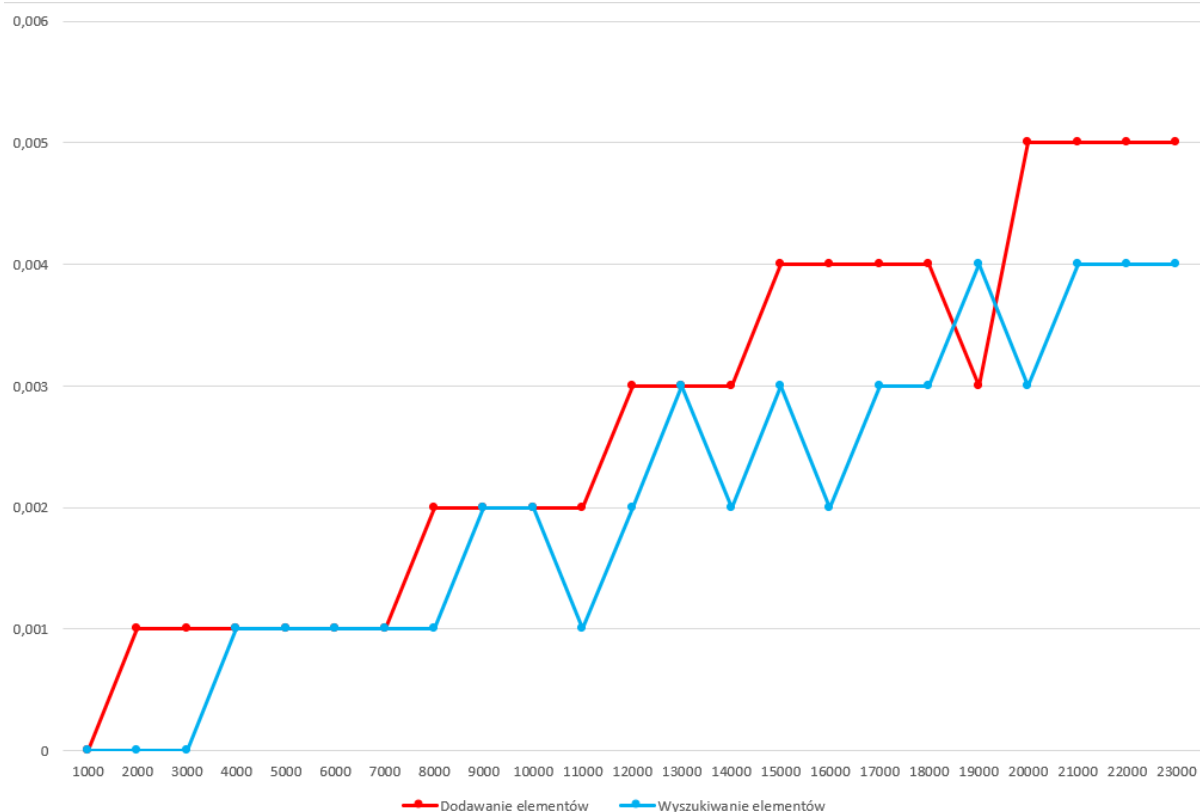
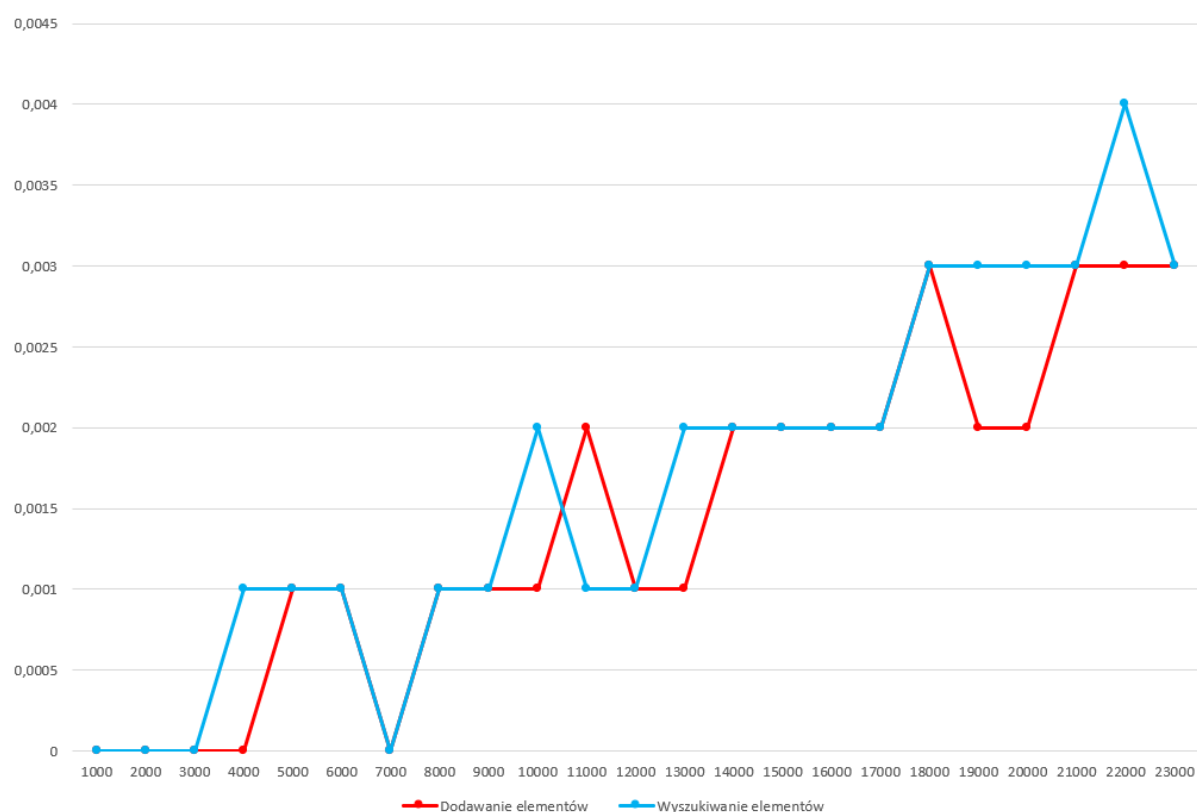


Tabela z czasami działania algorytmu dla elementów dodawanych w sposób
połowienia binarnego

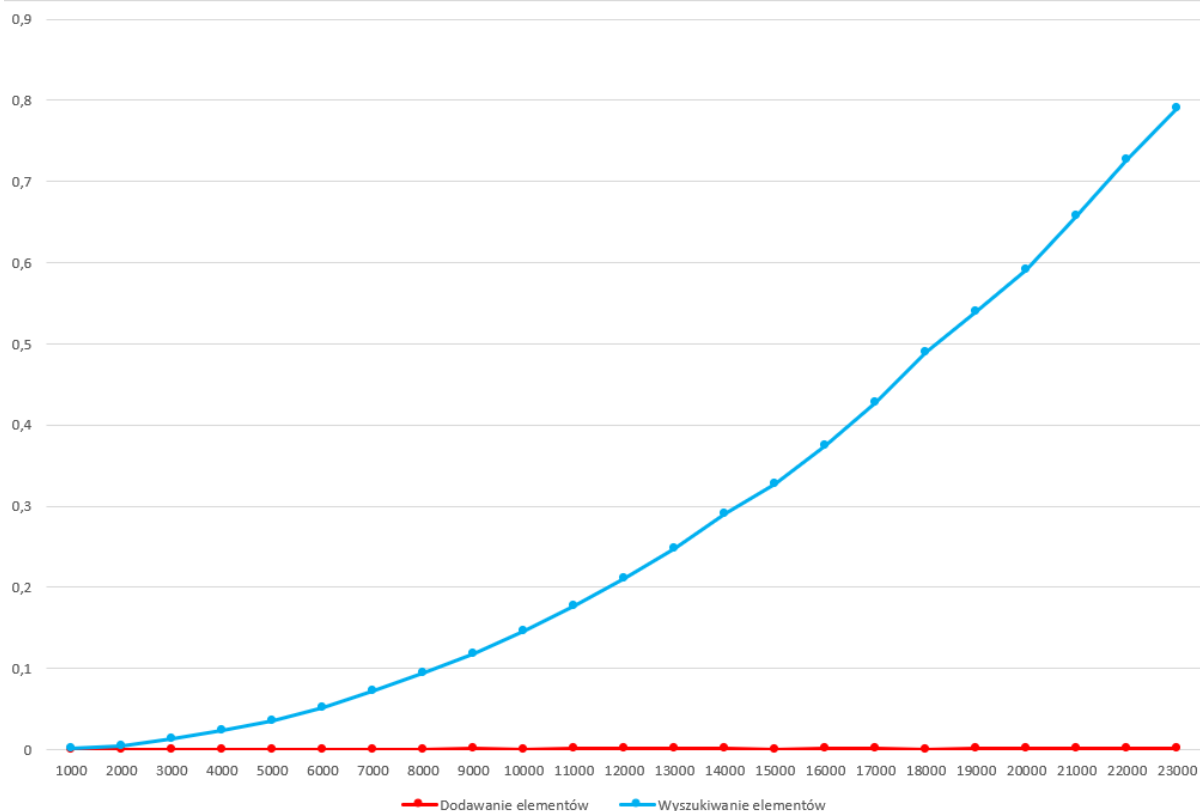
[liczba elementów]	Dodawanie elementów	Wyszukiwanie elementów
1000	0.000000	0.000000
2000	0.000000	0.000000
3000	0.000000	0.000000
4000	0.000000	0.001000
5000	0.001000	0.001000
6000	0.001000	0.001000
7000	0.000000	0.000000
8000	0.001000	0.001000
9000	0.001000	0.001000
10000	0.001000	0.002000
11000	0.002000	0.001000
12000	0.001000	0.001000
13000	0.001000	0.002000
14000	0.002000	0.002000
15000	0.002000	0.002000
16000	0.002000	0.002000
17000	0.002000	0.002000
18000	0.003000	0.003000
19000	0.002000	0.003000
20000	0.002000	0.003000
21000	0.003000	0.003000
22000	0.003000	0.004000
23000	0.003000	0.003000



Lista

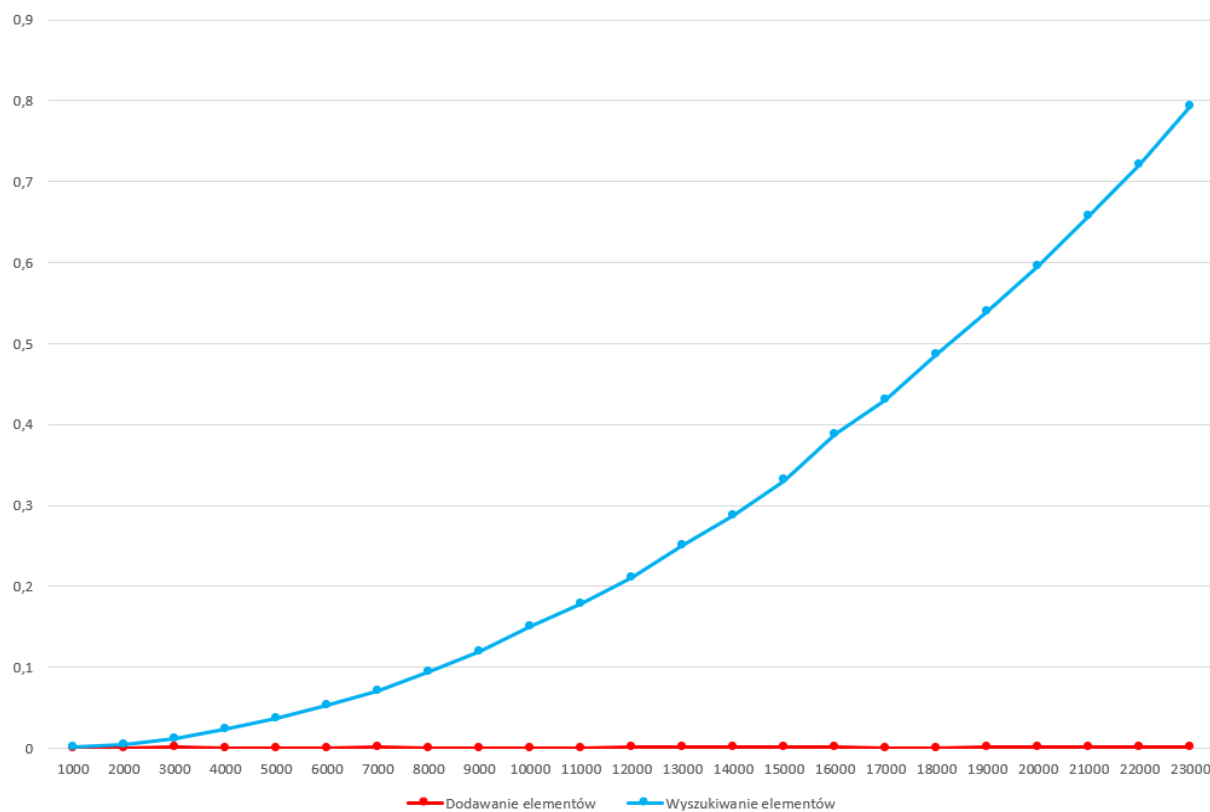
True – elementy dodawane w sposób losowy

[liczba elementów]	Dodawanie elementów	Wyszukiwanie elementów
1000	0.000000	0.001000
2000	0.000000	0.005000
3000	0.000000	0.013000
4000	0.000000	0.023000
5000	0.000000	0.036000
6000	0.000000	0.052000
7000	0.000000	0.072000
8000	0.000000	0.095000
9000	0.001000	0.118000
10000	0.000000	0.146000
11000	0.001000	0.177000
12000	0.001000	0.211000
13000	0.001000	0.247000
14000	0.001000	0.291000
15000	0.000000	0.328000
16000	0.001000	0.375000
17000	0.001000	0.427000
18000	0.000000	0.489000
19000	0.001000	0.539000
20000	0.001000	0.592000
21000	0.001000	0.657000
22000	0.001000	0.727000
23000	0.001000	0.791000



False – elementy dodawane nielosowo, w sposób rosnący

[liczba elementów]	Dodawanie elementów	Wyszukiwanie elementów
1000	0.000000	0.001000
2000	0.000000	0.005000
3000	0.001000	0.012000
4000	0.000000	0.023000
5000	0.000000	0.037000
6000	0.000000	0.053000
7000	0.001000	0.071000
8000	0.000000	0.094000
9000	0.000000	0.120000
10000	0.000000	0.150000
11000	0.000000	0.179000
12000	0.001000	0.211000
13000	0.001000	0.250000
14000	0.001000	0.288000
15000	0.001000	0.331000
16000	0.001000	0.388000
17000	0.000000	0.430000
18000	0.000000	0.486000
19000	0.001000	0.540000
20000	0.001000	0.595000
21000	0.001000	0.658000
22000	0.001000	0.721000
23000	0.001000	0.793000



Wnioski:

Porównanie czasów działania dla drzewa i listy:

W przypadku drzewa binarnego dla elementów ułożonych w sposób losowy, rosnąco i w sposób połowienia binarnego dla czasy działania algorytmów dla dodawania i wyszukiwania elementów są zbliżone.

Natomiast w przypadku listy algorytm jest bardziej wydajny dla dodawania elementów niż wyszukiwania.

Drzewo binarne:

Algorytm dla wyszukiwania i dodawania elementów jest najbardziej wydajny, gdy elementy są ułożone w sposób połowienia binarnego.

Algorytm jest najmniej wydajny dla elementów ułożonych rosnąco.

Lista:

Algorytm ma zbliżoną wydajność dla elementów ułożonych rosnąco i w sposób losowy.

Czas działania algorytmu dla elementów ułożonych rosnąco i w sposób losowy w przypadku dodawania elementów jest niemal identyczny i bardzo krótki. Czas wyszukiwania elementów również jest bardzo zbliżony.

Wnioski ogólne:

Najbardziej wydajnym z algorytmów jest drzewo binarne, jeżeli elementy są ułożone w odpowiedni sposób – rosnąco, lub w sposób połowienia binarnego. Wtedy czas działania algorytmu jest najkrótszy, zarówno dla wyszukiwania jak i dodawania elementów.