

Jędrzej Nowak 145164

Antoni Konecko 130528

Sprawozdanie nr 2

„Lista jednokierunkowa i drzewo BST”

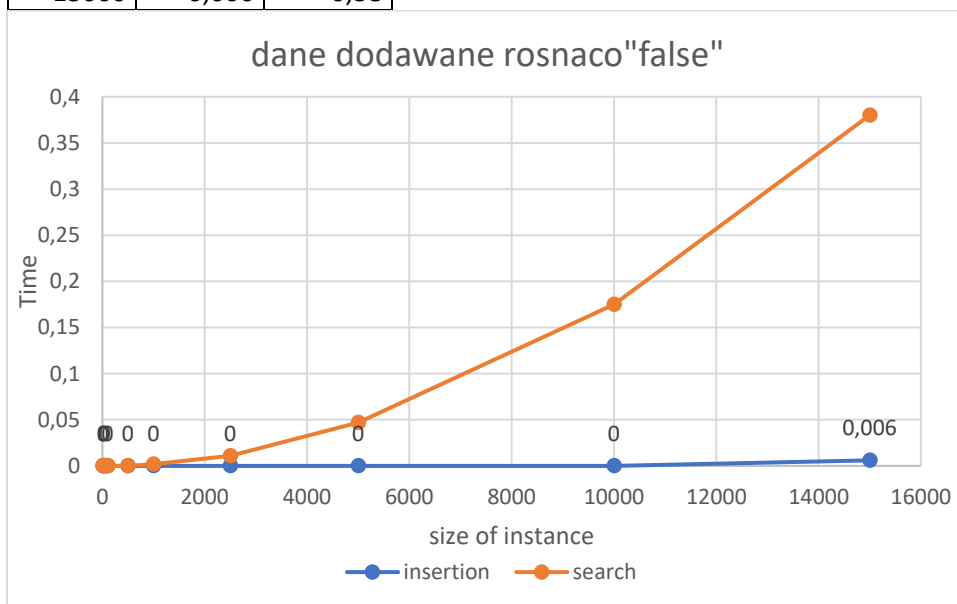
Lista jednokierunkowa

Wynik programu list.c

Size of instance	insertion	search	
10	0	0	false
20	0	0	false
50	0	0	false
100	0	0	false
500	0	0	false
1000	0	0,002	false
2500	0	0,011	false
5000	0	0,047	false
10000	0	0,175	false
15000	0,006	0,38	false
10	0	0	true
20	0	0	true
50	0	0	true
100	0	0	true
500	0	0	true
1000	0	0	true
2500	0	0	true
5000	0	0,031	true
10000	0	0,171	true
15000	0	0,375	true

Dane dodawane rosnąco (false)

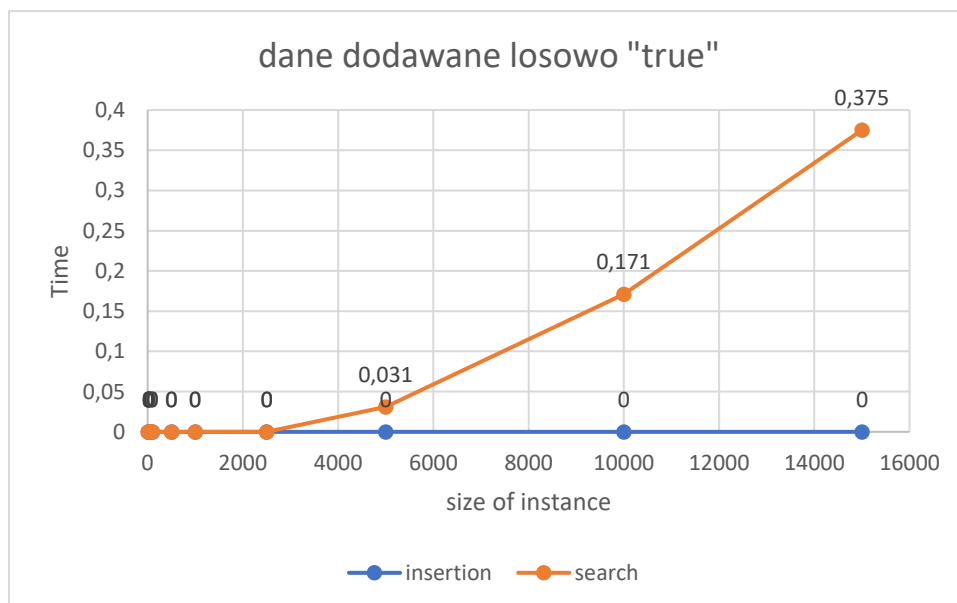
Size of instance	insertion	search
10	0	0
20	0	0
50	0	0
100	0	0
500	0	0
1000	0	0,002
2500	0	0,011
5000	0	0,047
10000	0	0,175
15000	0,006	0,38



Czas dodawania rosnąco elementów jest w granicy zera (jedynie dla $n=15\text{tys.}$ wynosi 0,006s), natomiast czas wyszukiwania rośnie wraz ze wzrostem wielkości instancji.

Dane dodawane losowo „true”

size of instance	insertion	search
10	0	0
20	0	0
50	0	0
100	0	0
500	0	0
1000	0	0
2500	0	0
5000	0	0,031
10000	0	0,171
15000	0	0,375



W przypadku losowego dodawania danych wynik jest zbliżony. Czas dodawania również wynosi ok. 0, a czas wyszukiwania rośnie ze wzrostem instancji, a wartości są zbliżone do tych z poprzedniego wykresu.

Drzewo BST

Dane wejściowe : 10, 20, 50, 100, 500, 1000, 2500, 5000, 10000, 15000

Średni czas dodawania elementu dla różnych sposobów rozkładu danych

Increasing = 0,137s

Random = 0,0056s

Binary = 0,0002s

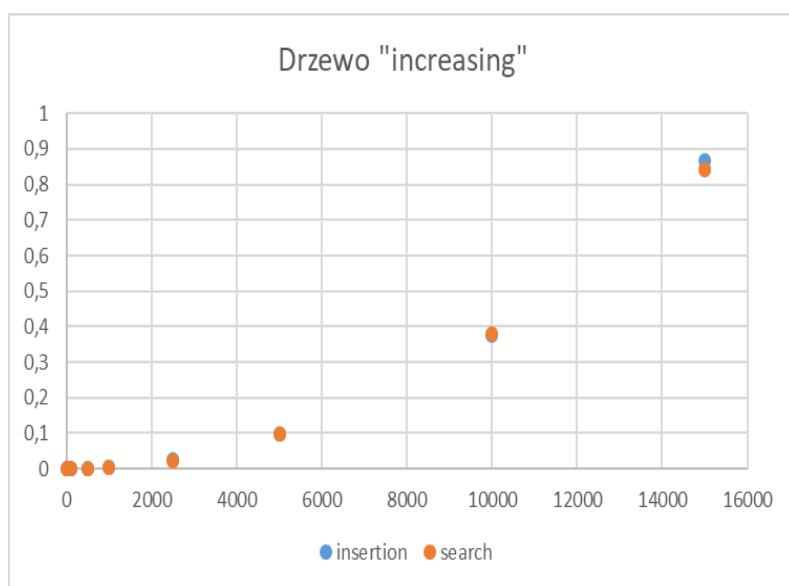
Średni czas wyszukiwania elementu dla różnych sposobów rozkładu danych

Increasing = 0,135s

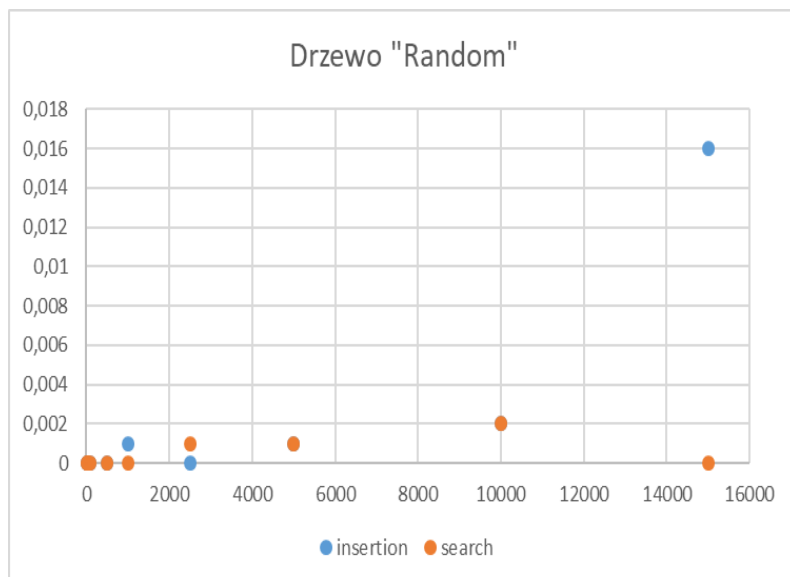
Random = 0,0004s

Binary = 0,0003s

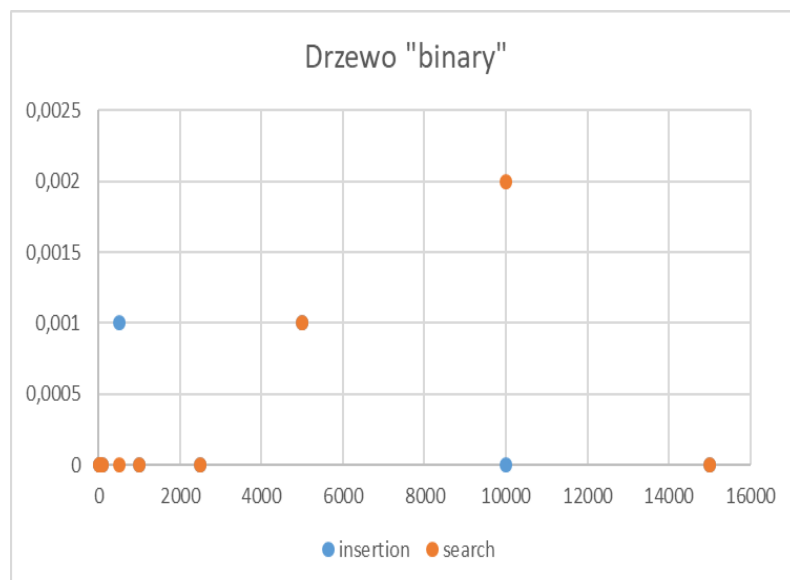
Input	insertion	search
10	0	0
20	0	0
50	0	0
100	0	0
500	0,001	0,001
1000	0,003	0,004
2500	0,024	0,023
5000	0,098	0,096
10000	0,376	0,381
15000	0,867	0,843



Input	insertion	search
10	0	0
20	0	0
50	0	0
100	0	0
500	0	0
1000	0,001	0
2500	0	0,001
5000	0,001	0,001
10000	0,002	0,002
15000	0,016	0



Input	insertion	search
10	0	0
20	0	0
50	0	0
100	0	0
500	0,001	0
1000	0	0
2500	0	0
5000	0,001	0,001
10000	0	0,002
15000	0	0



Wnioski: Na podstawie średnich czasów pracy oraz wykresów można wywnioskować, że najkorzystniejszym rozkładem danych dla drzewa jest sposób binarny, a tym najmniej korzystnym jest sposób w którym dane rozłożone są rosnąco