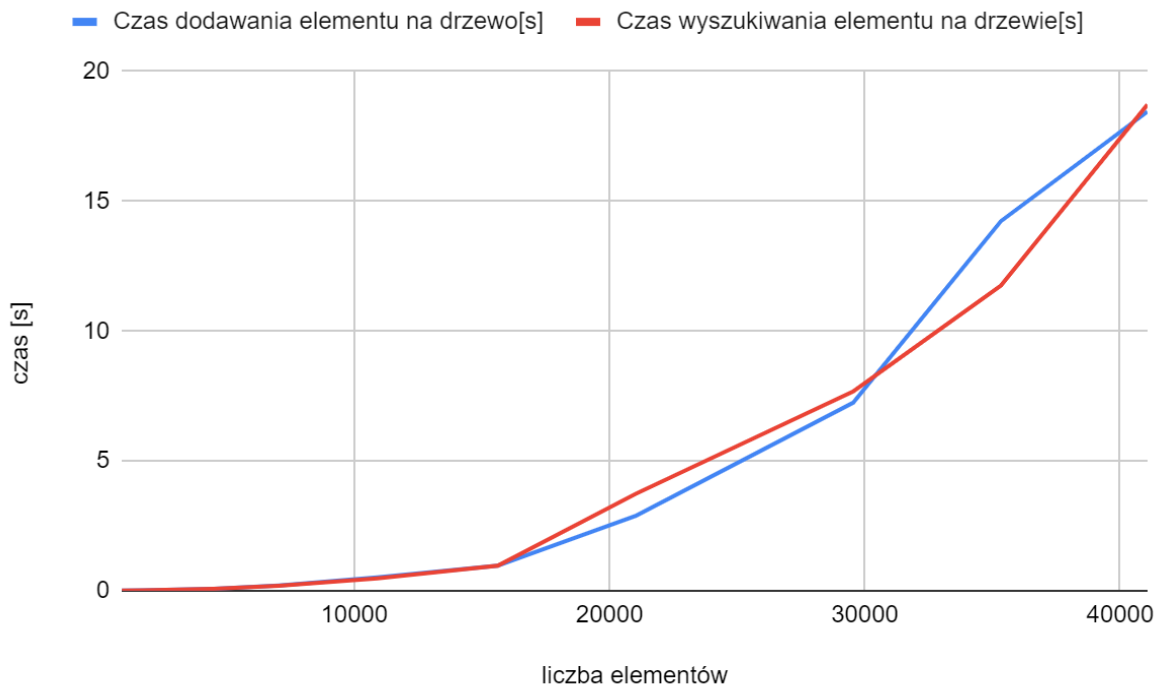


RAPORT- Binary Search Tree

Tabela oraz wykres pomiarów czasów dla elementów dodawanych w sposób wzrastający:

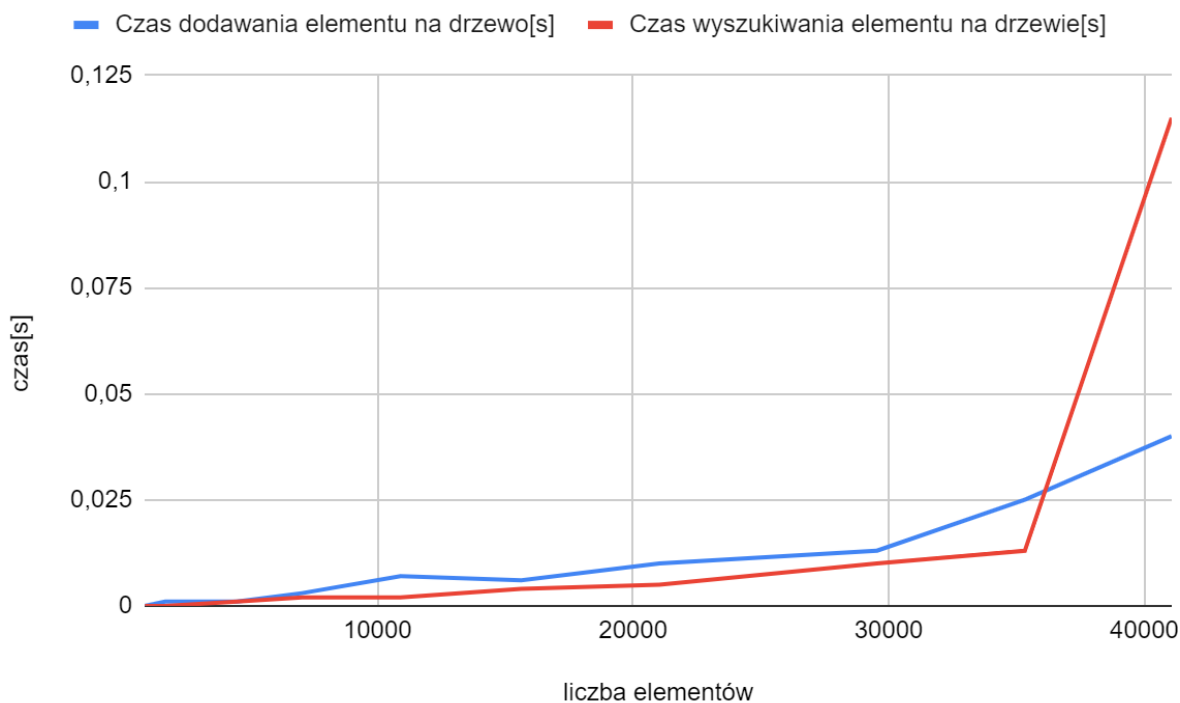
liczba elementów	Czas dodawania elementu na drzewo[s]	Czas wyszukiwania elementu na drzewie[s]
900	0,008	0,002
1700	0,011	0,011
4504	0,072	0,079
7054	0,205	0,188
10900	0,517	0,477
15620	0,968	0,966
21032	2,884	3,736
29543	7,242	7,676
35329	14,233	11,751
41069	18,452	18,724



Wnioski: Zarówno wyszukiwanie elementów na drzewie i ich dodawanie przebiega czasowo podobnie, z małą przewagą dodawania elementów, co jednakże ulega zmianie gdyż przy około 30 tysiącach elementów wyszukiwanie elementów na drzewie ułożonych wzrastająco staje się wydajniejsze czasowo.

Tabela oraz wykres pomiarów czasów dla elementów dodawanych w sposób losowy:

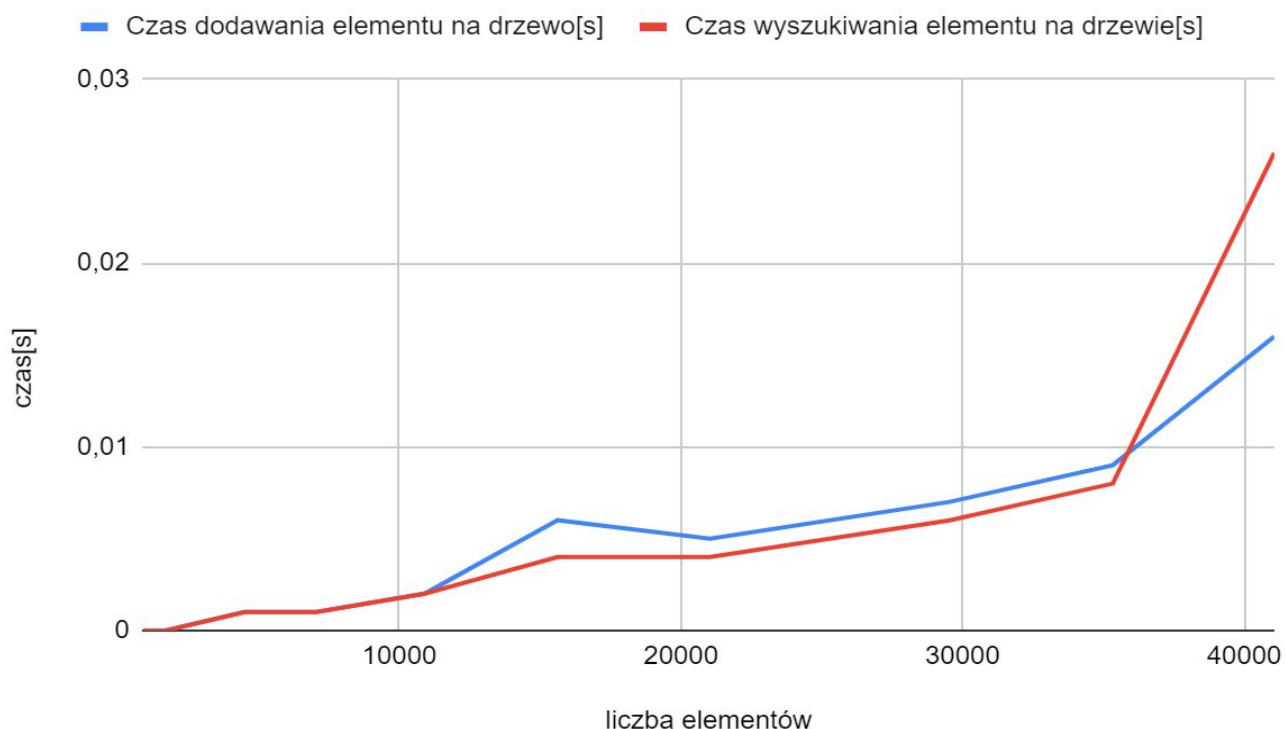
liczba elementów	Czas dodawania elementu na drzewo[s]	Czas wyszukiwania elementu na drzewie[s]
900	0	0
1700	0,001	0
4504	0,001	0,001
7054	0,003	0,002
10900	0,007	0,002
15620	0,006	0,004
21032	0,01	0,005
29543	0,013	0,01
35329	0,025	0,013
41069	0,04	0,115



Wnioski: Czas dodawania elementów losowych na drzewo binarne jest bardzo zbliżony czasowi wyszukiwania tych elementów. Łatwo jednak zauważyć, że przekraczając barierę ok. 35 tysięcy losowo ułożonych elementów na drzewie, ich wyszukiwanie trwa znacznie dłużej, podczas gdy czas dodawanie elementów rośnie raczej jednostajnie.

Tabela oraz wykres pomiarów czasów dla elementów dodawanych w sposób połowienia binarnego:

liczba elementów	Czas dodawania elementu na drzewo[s]	Czas wyszukiwania elementu na drzewie[s]
900	0	0
1700	0	0
4504	0,001	0,001
7054	0,001	0,001
10900	0,002	0,002
15620	0,006	0,004
21032	0,005	0,004
29543	0,007	0,006
35329	0,009	0,008
41069	0,016	0,026



Wnioski: Podobnie jak wyżej, dodawanie elementów w sposób połowienia binarnego jest niewiele wolniejsze niż ich wyszukiwanie, lecz po przekroczeniu bariery ok 35.tys elementów dodawanie elementów staje się wydajniejsze czasowo.

WNIOSKI CAŁOŚCIOWE: Dodawanie elementów na drzewo w sposób rosnący oraz wyszukiwanie tychże elementów jest najmniej wydajne czasowo. Kiedy liczba elementów przekracza 30 tys. czas ich implementowania jest bardzo długi. Drzewo implementowane za pomocą elementów randomowych zostaje zbudowane znacznie szybciej, również znacznie zwiększa się prędkość przeszukiwania jego elementów. Czasy implementowania drzewa w sposób połowienia binarnego są nieco krótsze od czasów implementowania drzewa z elementów randomowych, i są praktycznie zerowe. Dzieje się tak ponieważ metoda połowienia binarnego jest najbardziej optymalna i gwarantuje powstanie drzewa zrównoważonego w bardzo krótkim czasie.