

# Sprawozdanie z laboratorium nr 7

## „Złożoność obliczeniowa algorytmów w drzewie”

Filip Chodorowski

1 czerwca 2015

### Spis treści

<b>1</b>	<b>Założenia zadania</b>	<b>1</b>
<b>2</b>	<b>Zaimplementowane struktury</b>	<b>2</b>
2.1	Drzewo binarne . . . . .	2
2.2	Drzewo czerwono-czarne . . . . .	2
<b>3</b>	<b>Wyniki</b>	<b>3</b>
3.1	Porównanie dodawania elementów dla drzewa binarnego i drzewa czerwono-czarnego . . . . .	3
3.2	Porównanie wyszukiwania wszystkich elementów drzewa dla drzewa binarnego i czerwono-czarnego . . . . .	4
3.3	Porównanie wyszukiwania elementu nieznajdującego się w drzewach . . . . .	5
<b>4</b>	<b>Wnioski</b>	<b>6</b>

### 1 Założenia zadania

Zadanie polegało na zaimplementowaniu struktury drzewa binarnego i drzewa czerwono-czarnego z algorytmami dodawania i wyszukiwania elementów.

## 2 Zaimplementowane struktury

### 2.1 Drzewo binarne

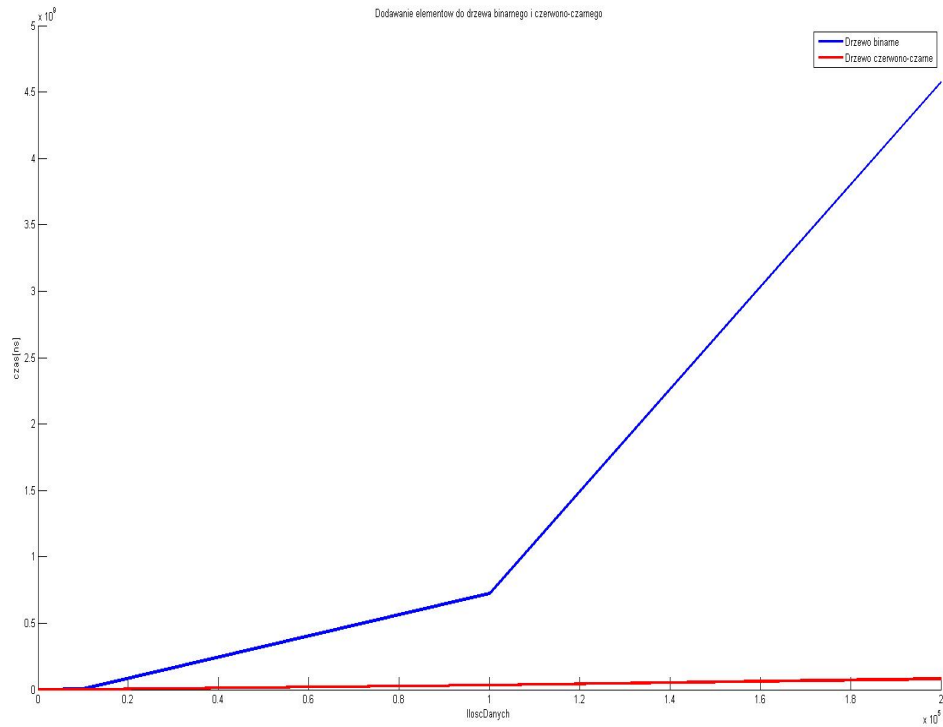
Drzewo binarne posiada co najwyżej 2 kolejne węzły. Jest strukturą niezbalansowaną tzn. wysokość drzewa może być dwukrotnie większa niż wysokość minimalna. Optymistyczna teoretyczna klasa złożoności podstawowych operacji:  $O(\log_2 n)$  W najgorszym przypadku drzewo binarne jest traktowane jak lista i złożoność podstawowych operacji wynosi wtedy:  $O(n)$

### 2.2 Drzewo czerwono-czarne

Drzewo czerwono-czarne jest odmianą drzewa binarnego, z tego powodu każdy węzeł posiada co najwyżej 2 kolejne. Jest strukturą zbalansowaną tzn. wysokość drzewa nie może być dwukrotnie większa niż wysokość minimalna. Ostatnie węzły są liśćmi, które wskazują adres węzła strażnika. Teoretyczna klasa złożoności podstawowych operacji:  $O(\log_2 n)$  Nie ma przypadku pesymistycznego, ponieważ drzewo jest zbalansowane.

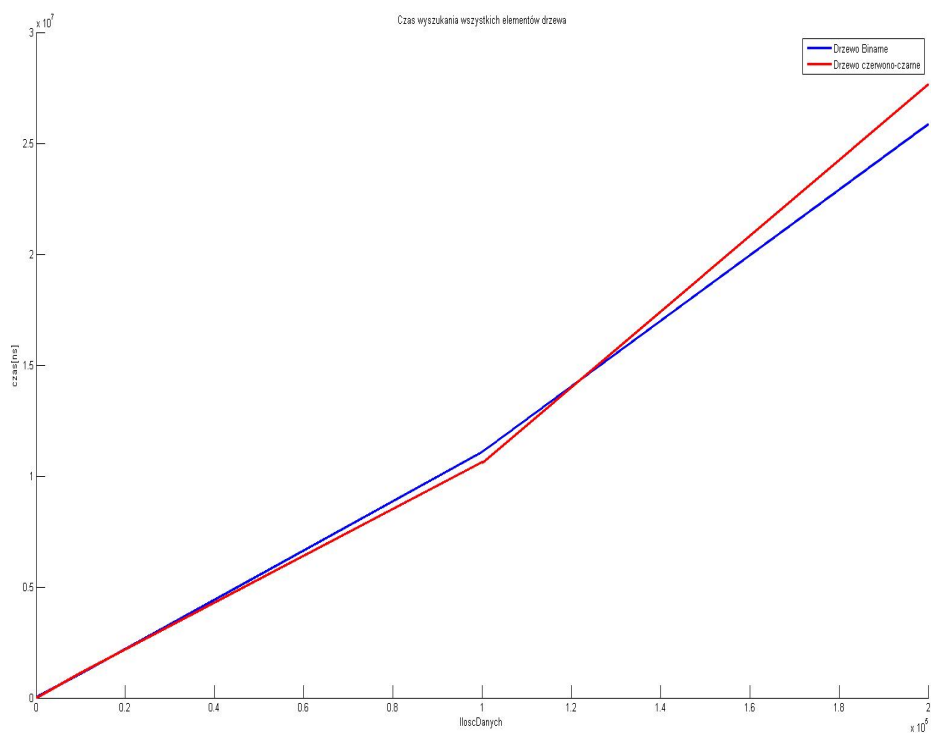
### 3 Wyniki

#### 3.1 Porównanie dodawania elementów dla drzewa binarnego i drzewa czerwono-czarnego



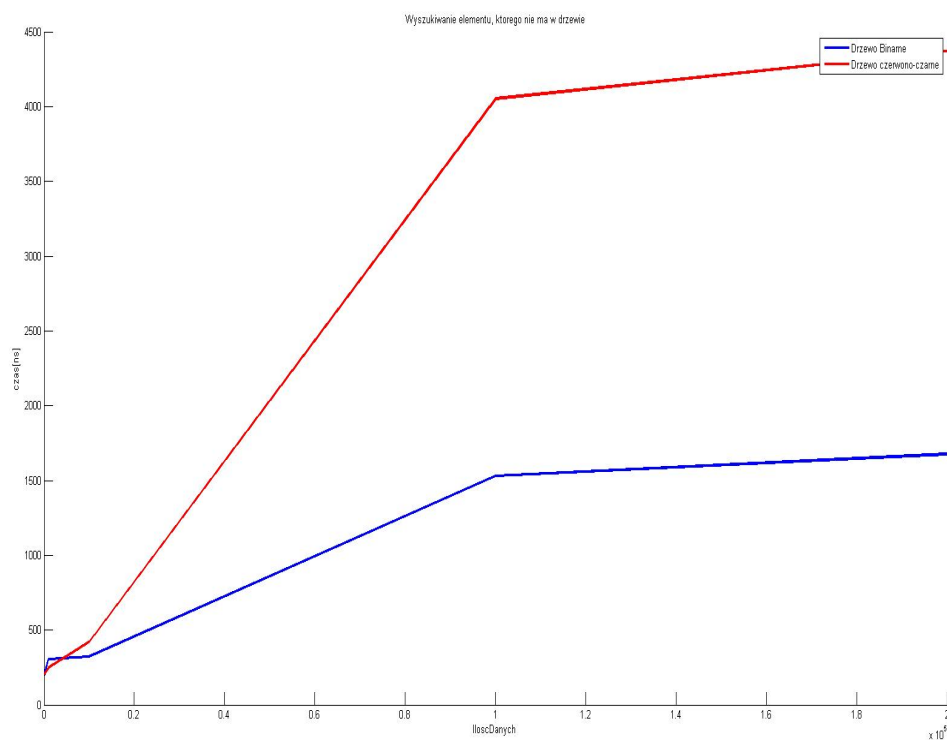
Drzewo binarne dodaje elementy szybciej dla rozmiaru 100, powyżej rozmiaru 100 drzewo czerwono-czarne jest szybsze, dla 200 000 elementów jest to różnica 100 razy i tendencja nadal rośnie. W obydwu drzewach elementy o takiej samej wartości są umieszczane w kolejnym niższym węźle. W drzewie binarnym struktura nie jest balansowana, co powoduje przy większych danych złożoność:  $O(n)$

### 3.2 Porównanie wyszukiwania wszystkich elementów drzewa dla drzewa binarnego i czerwono-czarnego



Czas wyszukania wszystkich elementów obydwu drzew jest tego samego rzędu. Wynika to z powtarzających się elementów w strukturach. Chociaż drzewo binarne nie jest zbalansowane to w tym konkretnym przypadku, odrzuca podczas wyszukiwania bardzo wiele elementów powtarzających się.

### 3.3 Porównanie wyszukiwania elementu nieznajdującego się w drzewach



Czas wyszukiwania nieistniejącego elementu w drzewie jest porównywalny dla. Dla drzewa binarnego czas wykonywania zależy od wyszukiwanego elementu, a dla drzewa czerwono-czarnego nie. Złożoność wyszukiwania drzewa czerwono-czarnego:  $O(\log_2 n)$

Złożoność wyszukiwania drzewa binarnego: Od  $O(\log_2 n)$  do  $O(n)$

## 4 Wnioski

- Drzewo czerwono-czarne ma stałą złożoność podstawowych operacji wynoszącą:  $O(\log_2 n)$
- Nowe elementy o tej samej wartości w drzewie binarnym (jeśli muszą być umieszczane) lepiej umieszczać w wyższych wierzchołkach, bo inaczej trzeba wykonać dodatkowe operacje przejścia na dół struktury.
- drzewo czerwono-czarne  $>$  binarne, chyba że mamy wcześniej posortowane elementy, wtedy można wykorzystać binarne bez obawy o złożoność  $O(n)$