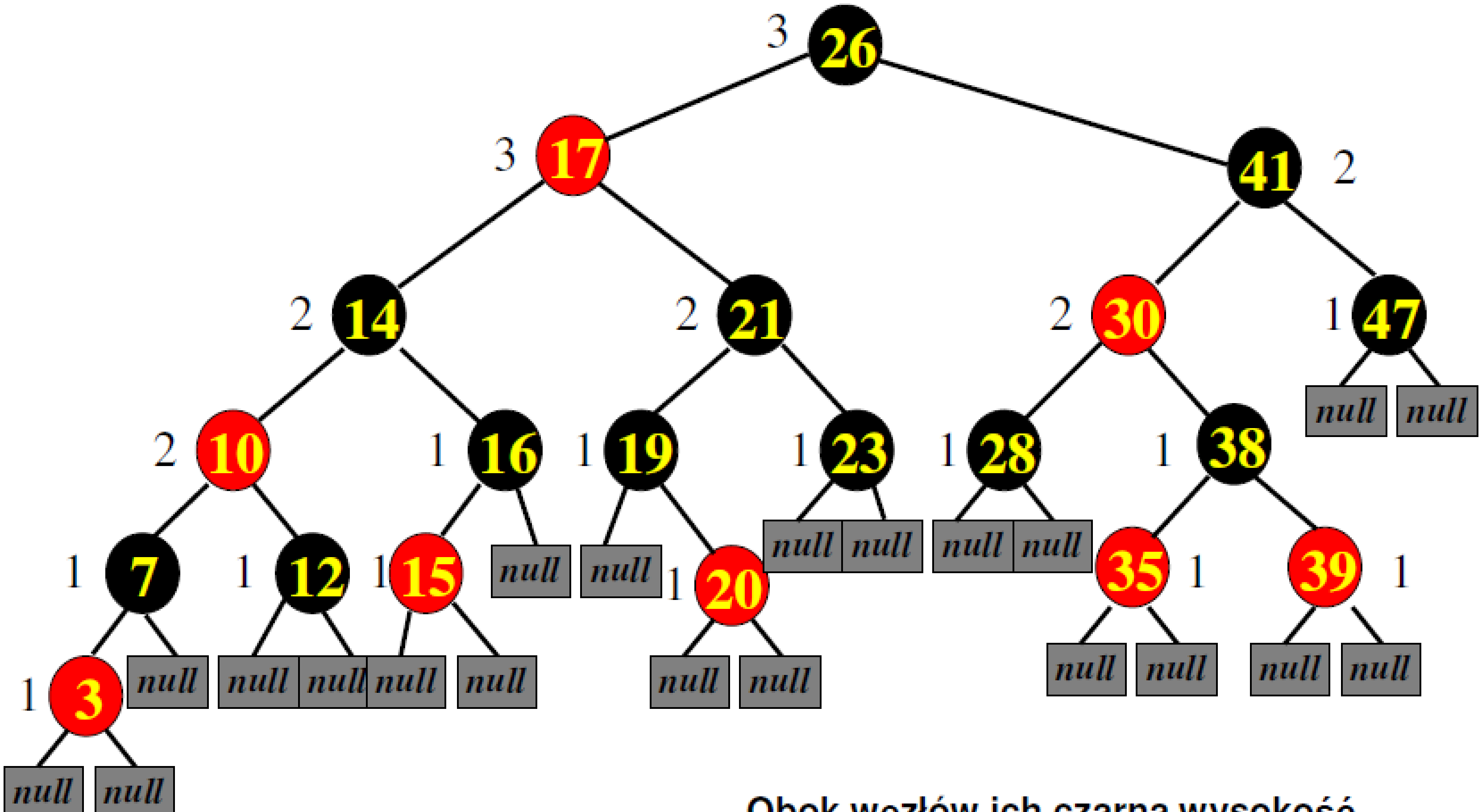
SDIZO-BST

Drzewem czerwono-czarnym (RB tree) nazywany drzewo poszukiwań **binarnych, dla którego każdy węzeł posiada dodatkowy bit koloru (albo czerwony albo czarny**) **o następujących własnościach:**

1. Każdy węzeł posiada kolor – czerwony lub czarny.
2. Korzeń jest koloru czarnego.
3. Każdy liść (*null*) jest czarny.
4. Dla czerwonego węzła obydwoje dzieci są czarne.
5. Każda ścieżka od ustalonego węzła do liścia musi zawierać tą samą ilość czarnych węzłów.

Definicja: Czarną wysokością węzła– *bh*(*x*), nazywamy ilość czarnych węzłów na ścieżce prowadzącej od tego węzła do dowolnego liścia.

SDIZO-BST



Obok węzłów ich czarna wysokość
(węzły *null* mają czarną wysokość 0)

WSTAWIANIE

Nowy węzeł wstawiamy tak, jak do zwykłego drzewa BST i kolorujemy na czerwono.

Jeśli własności drzewa RB zostały zaburzone, to naprawiamy drzewo poprzez zamianę kolorów i odpowiednie rotacje.

Które z własności mogą zostać zaburzone?

1. Każdy węzeł jest czerwony lub czarny - NIE
2. Korzeń jest czarny - TAK
3. Każdy liść (*null*) jest czarny - NIE
4. Obydwaj potomkowie czerwonego węzła są czarni - TAK
5. Każda ścieżka od węzła do liścia ma tą samą ilość czarnych węzłów - NIE

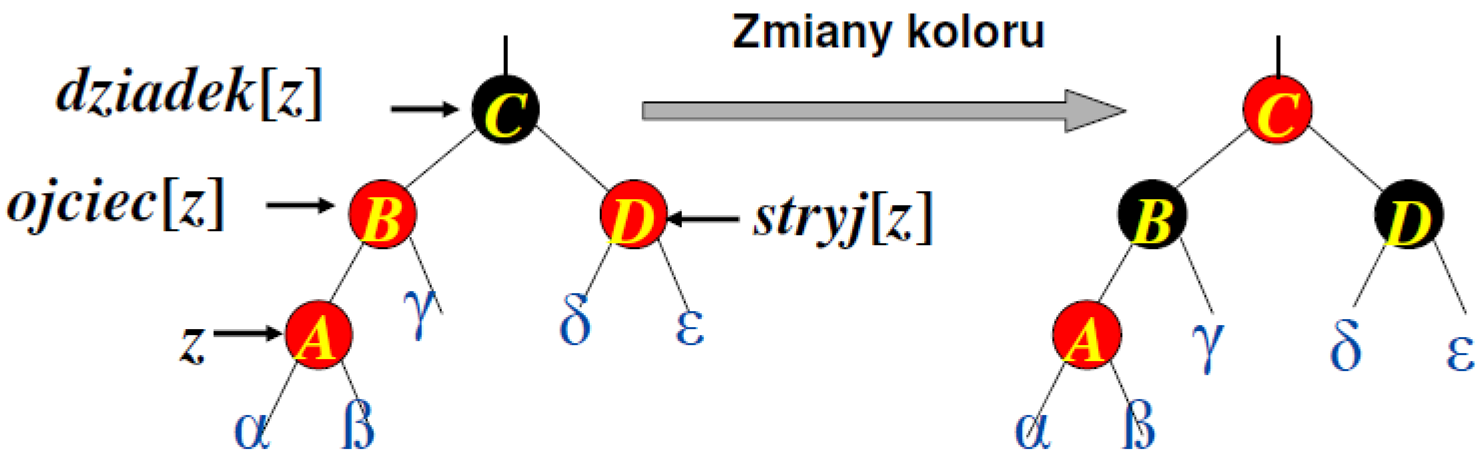
KOREKCJA

2.Korzeń jest czarny – wstawiany element jest korzeniem – kolorujemy go na czarno

4.Obydwaj potomkowie czerwonego węzła są czarni – ojciec jest czerwony a nasz wstawiany element zgodnie z algorytmem również

a) stryj wstawianego węzła (brat ojca) jest czerwony
b) stryj jest czarny, a wstawiany węzeł jest lewym potomkiem
c) stryj jest czarny, a wstawiany węzeł jest prawym potomkiem

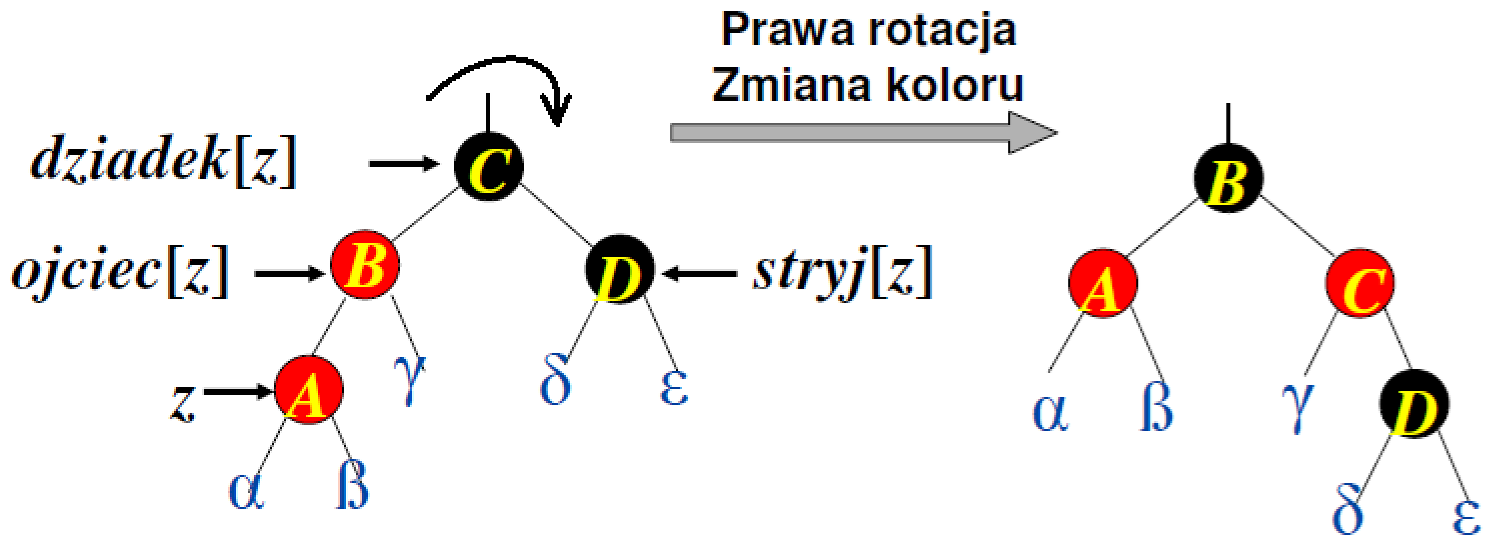
AD. A Stryj wstawianego węzła ***z*** (brat ojca) jest czerwony – zmieniamy kolor ojca i stryja na czarny a dziadka na czerwony



- jeżeli C jest nowym korzeniem – kolorujemy go na czarno
- jeżeli dziadek C dalej zaburza własności drzewa – naprawiamy dalej drzewo (w C)

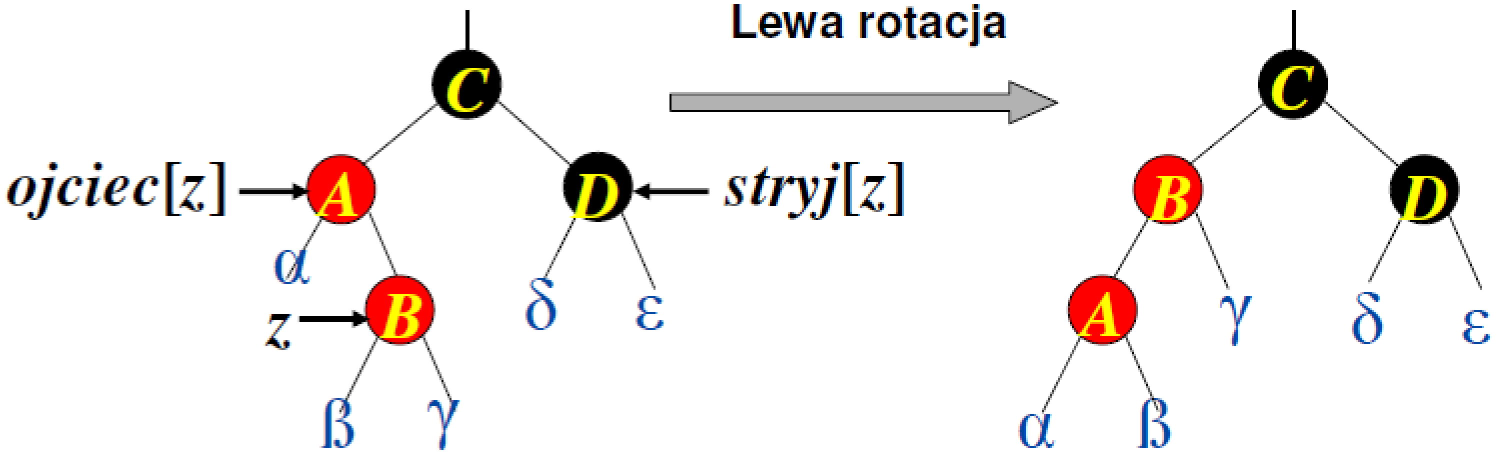
UWAGA! Przedstawione na tym slajdzie i następnych sytuacje są ogólne – tzn. każdy wierzchołek A,B,C,D może mieć poddrzewa

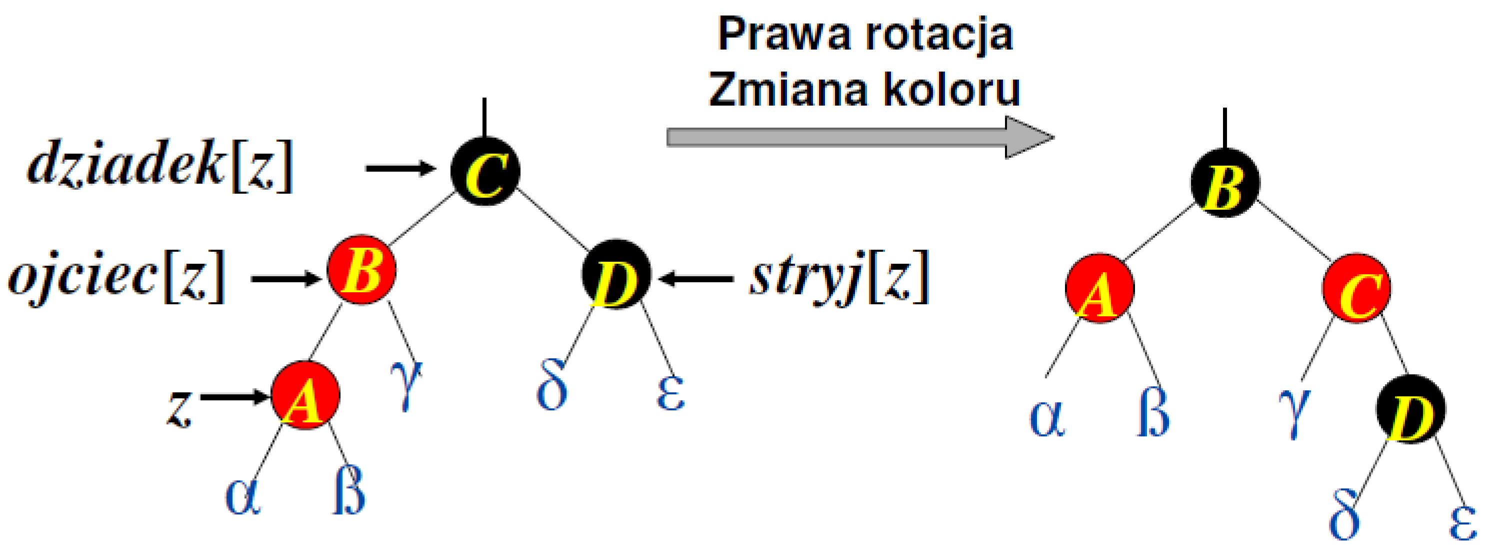
AD. B Stryj jest czarny, a wstawiany węzeł z jest lewym potomkiem - przeprowadzamy prawa rotacje w C i zmieniamy kolor *B* i *C*



Po tej operacji drzewo jest niezaburzone

AD. C Stryj jest czarny, a wstawiany węzeł jest prawym potomkiem - przeprowadzamy rotacje lewą dla A i mamy przypadek B





USUWANIE
Korzystamy ze standardowej operacji usuwania dla BST.
Jeżeli zaburzymy własności drzewa RB, to naprawiamy je wykorzystując rotacje i zmiany kolorów węzłów.

Jakie własności drzewa mogą zostać zaburzone?

1. Każdy węzeł ma kolor czarny lub czerwony - NIE
2. Korzeń jest czarny - TAK
3. Kady liść (*null*) jest czarny NIE
4. Obaj potomkowie czerwonego węzła są czarni - TAK
5. Każda ścieżka od wybranego węzła do liścia zawiera tyle samo węzłów czarnych TAK

Operacja usuwania jest bardziej skomplikowana – wymaga rozpatrzenia więcej przypadków i nie będzie tutaj przedstawiona

PODSUMOWANIE

- **drzewo czerwono-czarne o n wewnętrznych węzłach ma wysokość co najwyżej $2\log_2(n+1)$**
- **operacje modyfikacji drzewa (dodawanie i usuwanie) mają złożoność O(log n)**

http://eduinf.waw.pl/inf/alg/001_search/index.php

SDIZO-BST

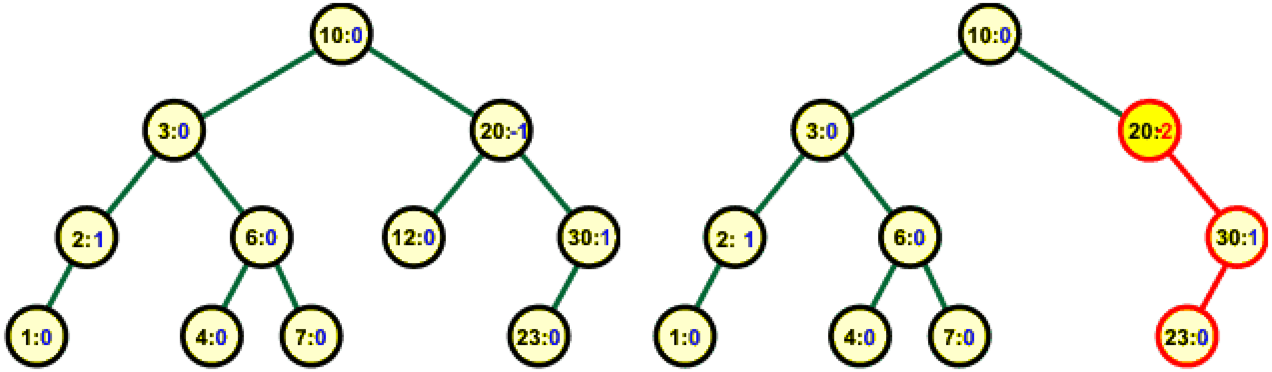
Drzewa AVL -drzewo BST zwane również samo-równoważącym się (samo – balansującym się). Nazwa pochodzi od nazwisk twórców - rosyjskich matematyków: Adelsona-Velskiego oraz Landisa

Zrównoważenie drzewa osiąga się, przypisując każdemu węzłowi współczynnik wyważenia, który jest równy różnicy wysokości lewego i prawego poddrzewa ($bf = H_L - H_R$). Może wynosić 0, +1 lub -1. Wstawiając lub usuwając elementy drzewa (tak, by zachować własności drzewa BST), modyfikuje się też współczynnik wyważenia, a gdy przyjmie on niedozwoloną wartość, wykonuje specjalną operację rotacji węzłów, która przywraca zrównoważenie.

Koszt modyfikacji drzewa jest nieco większy niż dla zwykłego BST, ale za to własności drzewa AVL gwarantują, że pesymistyczny czas wyszukiwania elementu w drzewie o n węzłach wynosi $1.44(\log_2 n)$ (dokładanie $1.44\log_2(n+2)-0.328$)

Drzewa AVL –przykład

opis węzła: pierwsza liczba to klucz, druga (po dwukropku) o współczynnik wyważenia węzła (liście mają 0)



drzewo po lewej jest drzewem AVL, po prawej nie

Drzewa AVL - Wstawianie

- wstawiamy jak do zwykłego BST,
- wykonujemy aktualizacje wyważeń węzłów od wstawionego elementu do korzenia. Jeżeli w danym węźle współczynnik wyważenia przyjął wartość -2 lub 2 to należy przywrócić właściwości drzewa rotacjami (max. potrzebne będą dwie rotacje)

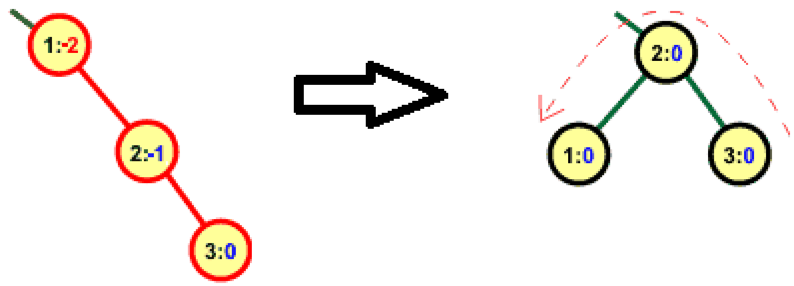
Usuwanie

- usuwany element tak jak w zwykłym BST
- wykonujemy aktualizacje wyważeń węzłów od rodzica usuniętego elementu aż do korzenia. Jeżeli w danym węźle współczynnik wyważenia przyjął wartość -2 lub 2 to należy przywrócić właściwości drzewa rotacjami. W najgorszym rotacje będą wykonywane od rodzica do korzenia

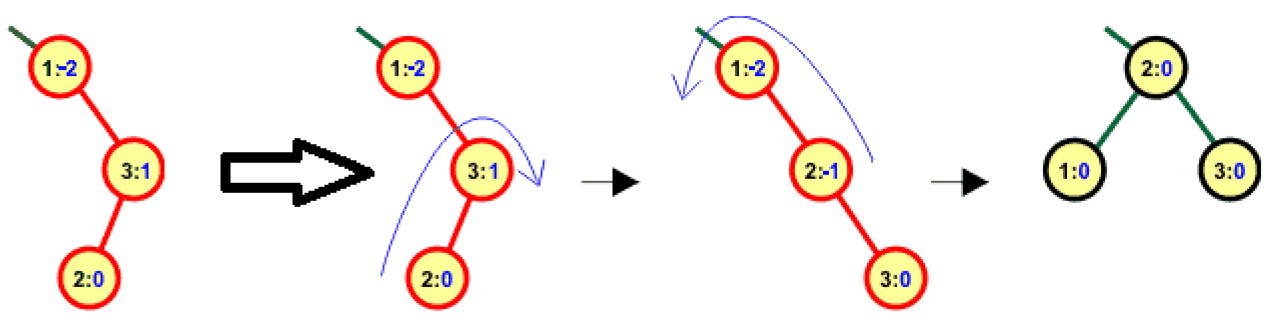
Złożoność operacji dodawania i usuwania - O(log*n*)

SDIZO-BST

Rotacje – są 4 typy, dwie pojedyncze i dwie podwójne



Rotacja pojedyncza



Rotacja podwójna