Zaimplementowana struktura drzewiasta: Drzewo czerwono - czarne

Własności drzewa:

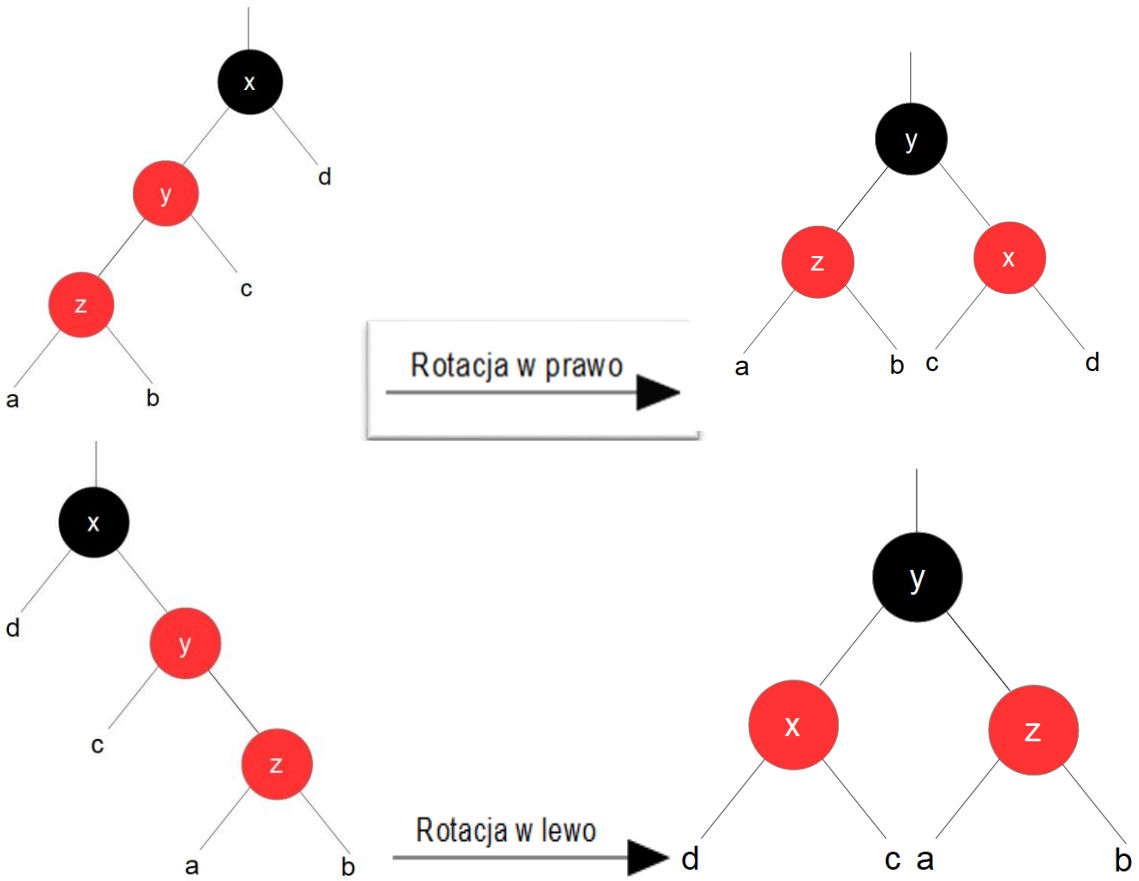
- korzeń jest czarny,
- liście (null) są czarne,
- czerwony węzeł nie może posiadać czerwonych dzieci,
- ścieżka z ustalonego węzła do liści musi liczyć tyle samo czarnych węzłów.

Definicja węzła drzewa:

- **bool red:** zmienna opisująca czy dany węzeł jest czerwony(true), lub czarny(false);
- **string data[3]:** tablica zawierająca imię([0]), nazwisko([1]) oraz adres([2]) abonenta;
- **list *head:** wskaźnik na początek listy zawierającej numery telefonu;
- **struct node *link[2]:** tablica wskaźników na lewe([0]) i prawe ([1]) dziecko.

Rotacje:

Pojedyncze:



Pseudokod:

1. Rotacja w prawo

x->red = true

y->red = fasle

x->link[0] = y->link[1]

y->link[1] = x

return y

2. Rotacja w lewo

x->red = true

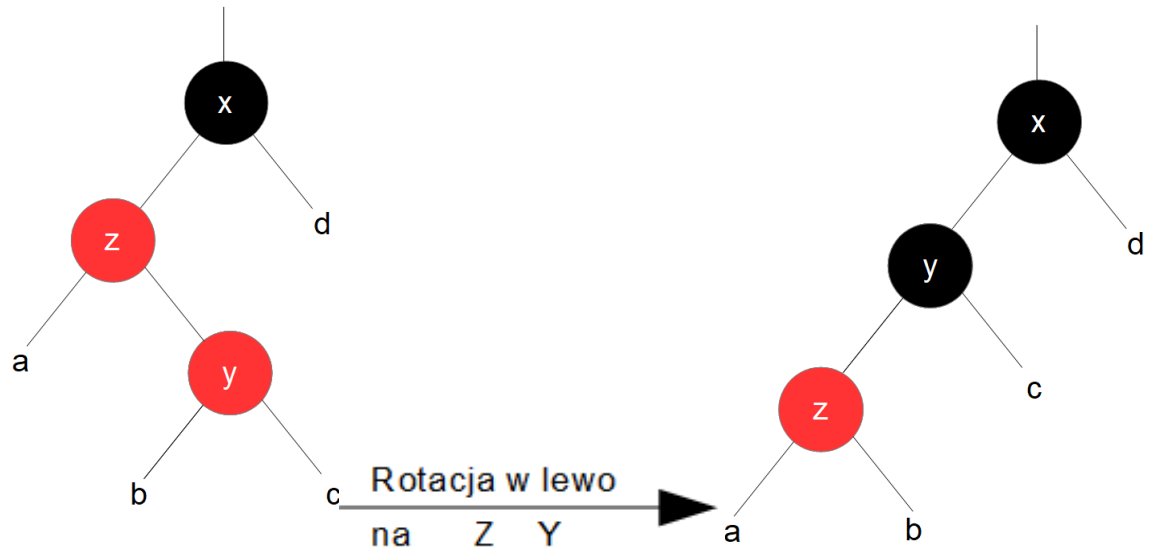
y->red = fasle

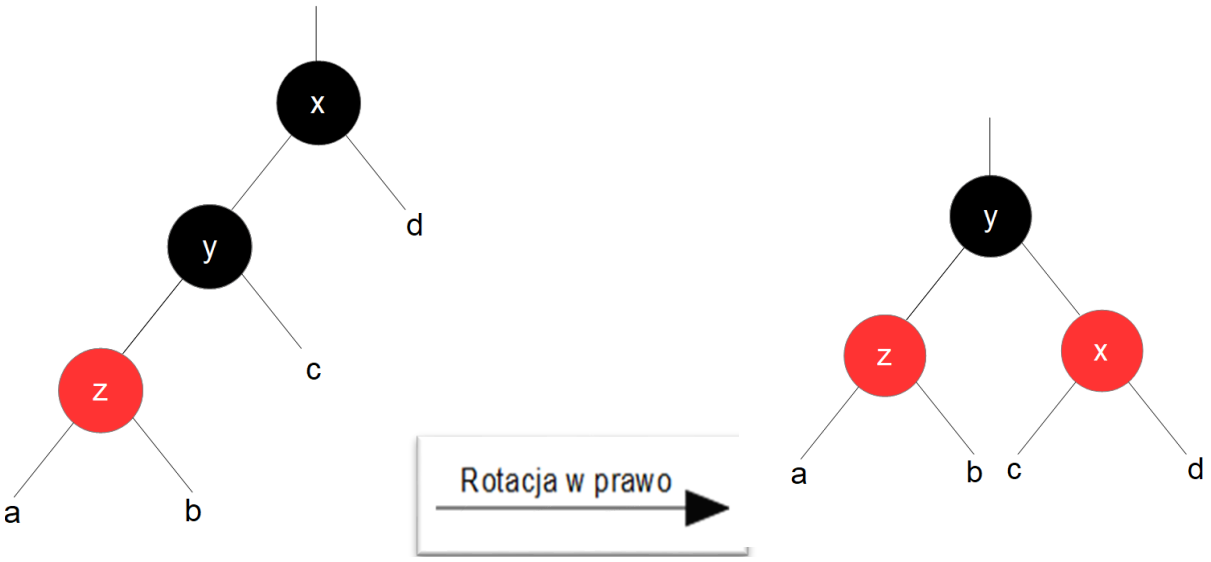
x->link[1] = y->link[0]

y->link[0] = x

return y

Podwójna:





Istnieje też lustrzany przypadek prawo - lewo, jest on analogiczny do przedstawionego powyżej.

Pseudokod:

z->red = true

y->red = fasle

z->link[1] = y->link[0]

y->link[0] = z

x->red = true

y->red = fasle

x->link[0] = y->link[1]

y->link[1] = x

return y

Dodawanie elementów:

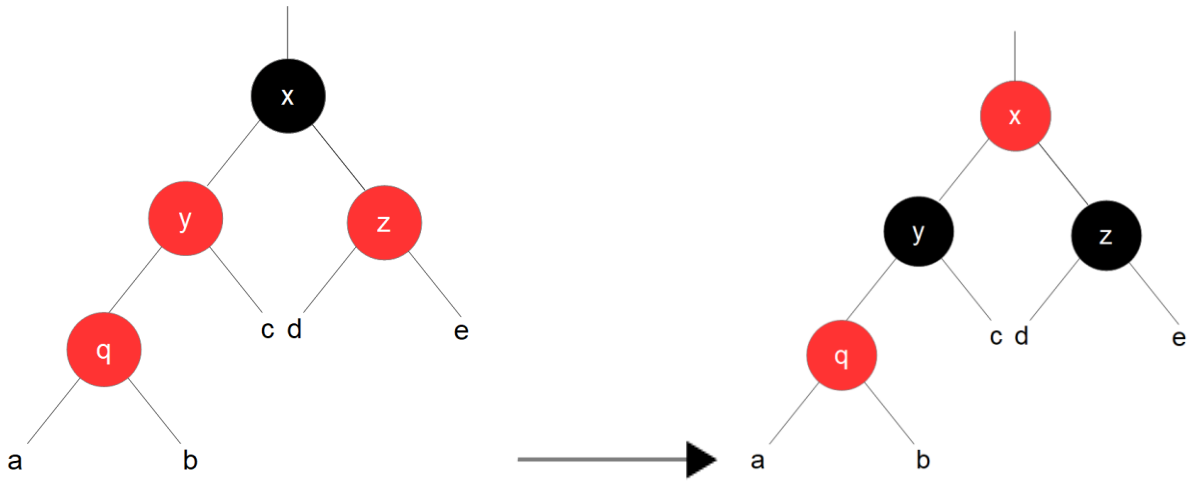
Dla prostoty, każdy wstawiany element będzie czerwony, gdyż wstawianie czerwonego elementu może w ogóle nie złamać zasad działania drzewa, a gdy już je złamie, to zawsze jest to zasada, że czerwony węzeł nie może mieć czerwonych synów. Jest to prostszy błąd do naprawienia niż błąd przy wstawianiu węzła czarnego, który nie dość, że zawsze będzie łamał zasadę o długości ścieżki, to na dodatek jest to błąd trudniejszy do naprawienia. Idziemy od korzenia w dół drzewa zwracając uwagę na 4 przypadki opisane poniżej.

Wstawianie elementu to nic szczególnie trudnego, po prostu tworzymy nowy węzeł, wypełniamy go zadanymi danymi, linki do synów ustawiamy na NULL, a jego kolor na czerwony. Jeżeli jest to pierwszy dodawany element staje się on korzeniem. Jeżeli zaś były już jakieś elementy to po prostu doczepiamy go pod dany wskaźnik.

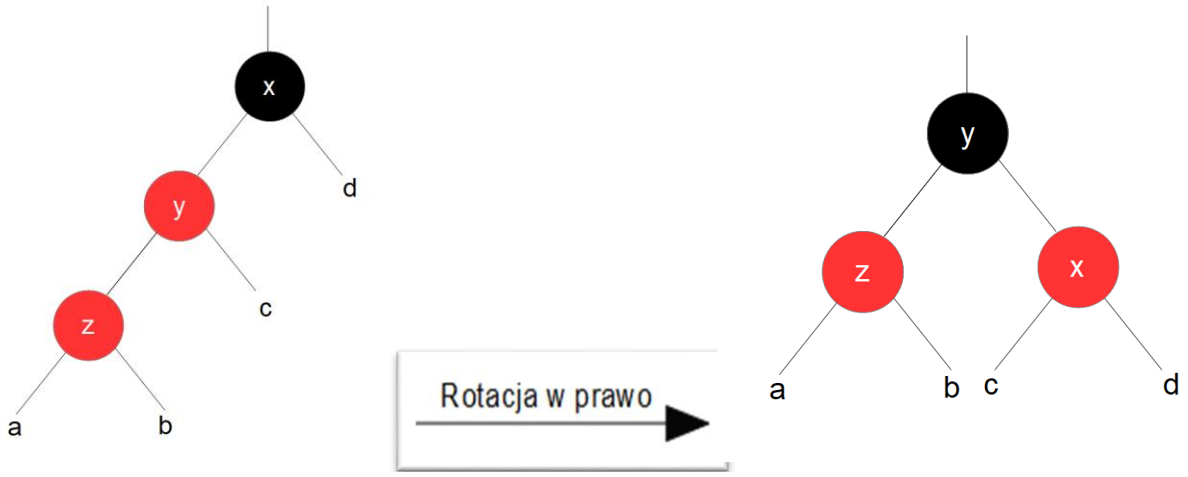
Idąc w dół drzewa i natrafiając na jedno z 3 przypadków wykonujemy albo przekolorowanie albo rotacje:

1. Podłączyliśmy węzeł pod węzeł czarny, zatem wszystko gra i śmiga, można więc zakończyć proces dodawania elementu.
2. Jeżeli napotkamy po drodze czarny węzeł, który ma czerwonych synów robimy rekoloryzacje, jeżeli pogwałciło to zasady drzewa (czyli wystąpiły 2 czerwone węzły po sobie robimy rotacje) przypadek 3 oraz 4.
3. Przekoloryzowaliśmy węzeł i jego ojciec jest węzłem czerwonym i jesteśmy jego tym samym dzieckiem co on, to znaczy on jest lewym(prawym) i my jesteśmy lewym(prawym), robimy zatem pojedynczą rotacje.
4. Przekoloryzowaliśmy węzeł i jego ojciec jest węzłem czerwonym i jesteśmy jego przeciwnym dzieckiem co on, to znaczy on jest lewym(prawym), a my prawym(lewym), robimy zatem podwójną rotacje.

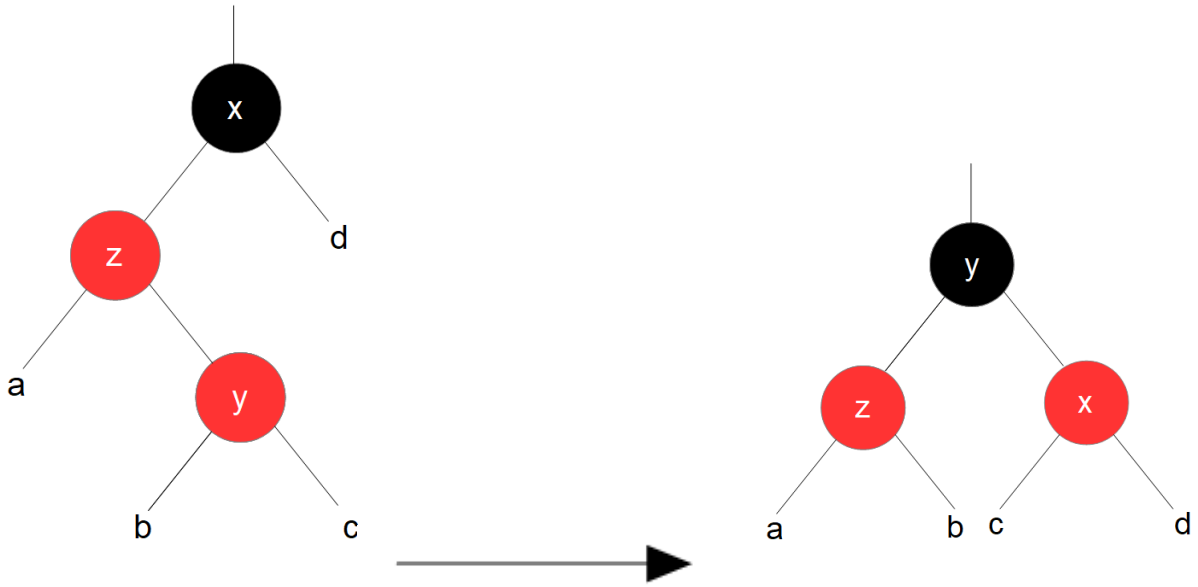
Przypadek 2:



Przypadek 3:



Przypadek 4:



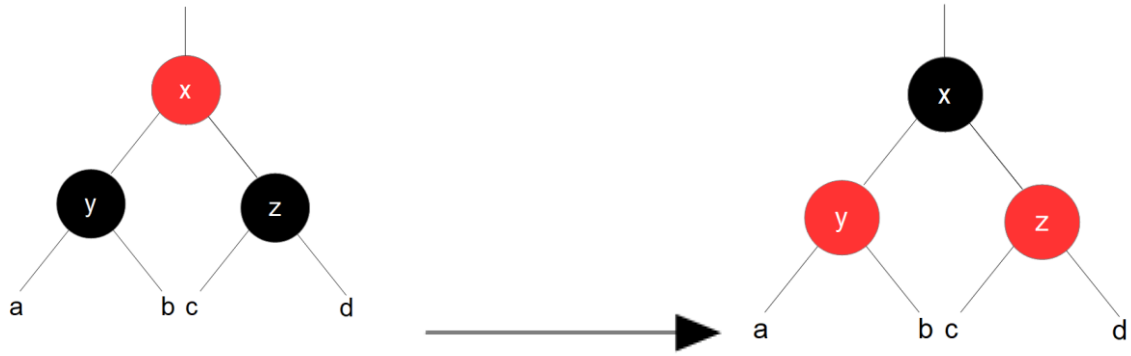
Usuwanie elementów:

Usuwanie będziemy robić metodą “pchania” czerwonego węzła w dół drzewa. Będziemy rekolorować drzewo w taki sposób, że usuwany węzeł będzie zastąpiony poprzednikiem który będzie czerwony, dzięki czemu po jego zniknięciu balans drzewa nie zostanie w żaden sposób zachwiany.

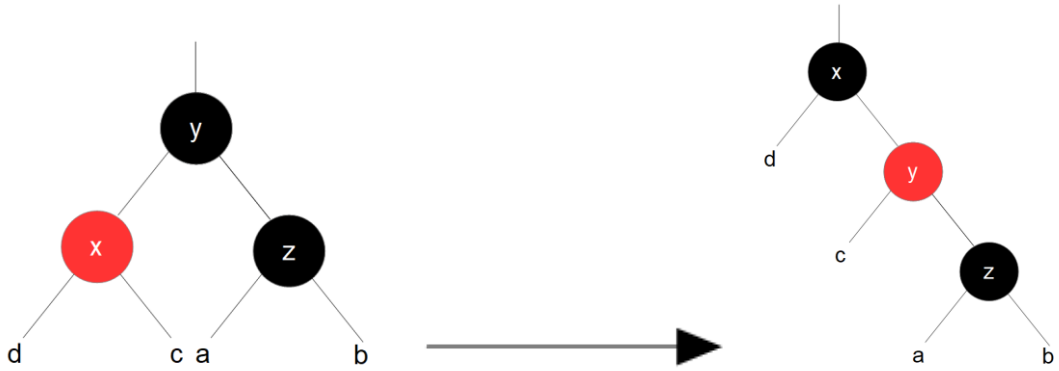
Rekoloryzacja drzewa ma 4 przypadki na które trzeba zwrócić uwagę.

1. Jeżeli węzeł oraz jego dzieci są czarne, oraz czterej wnukowie są czarni, to kolorujemy ojca na czarno a jego dzieci na czerwono.
2. Jeżeli węzeł ma jedno dziecko czarne, a drugie czerwone, robimy pojedynczą rotację.
3. Jeżeli węzeł ma dwójkę czarnych synów oraz wnuka czerwonego ze strony przeciwnej robimy podwójną rotację.
4. Jeżeli węzeł ma dwójkę czarnych synów oraz wnuka czerwonego z tej samej strony robimy pojedynczą rotację.

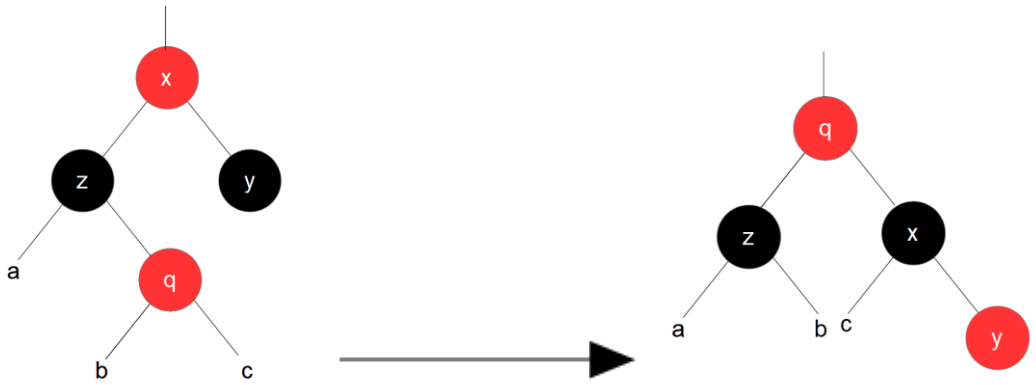
Przypadek 1:



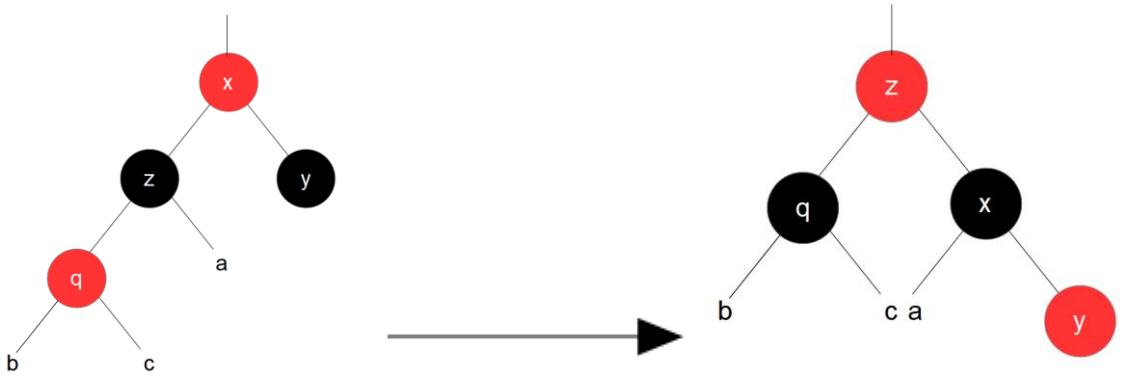
Przypadek 2:



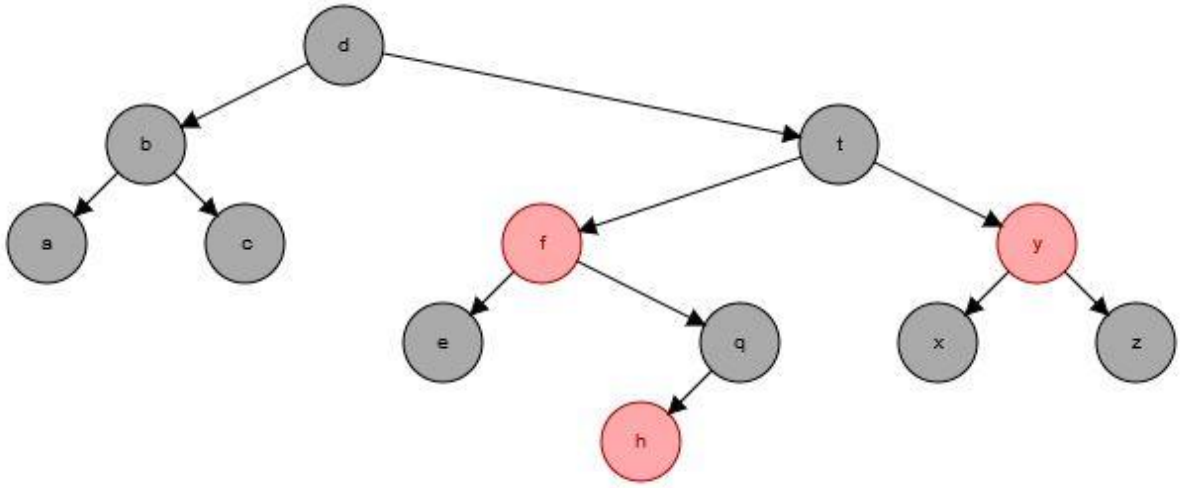
Przypadek 3:



Przypadek 4:



Drzewo przed usunięciem (usuwamy t):



Drzewo po usunięciu: