ALGORYTMY I STRUKTURY DANYCH

Temat 7:

Drzewa czerwono-czarne

Wykładowca: dr inż. Zbigniew TARAPATA

e-mail: Zbigniew.Tarapata@isi.wat.edu.pl

http://www.tarapata.strefa.pl/p_algorytmy_i_struktury_danych/

Współautorami wykładu są: A.Najgebauer, D.Pierzchała

Wykład : Algorytmy przetwarzania struktur drzewiastych

- drzewa czerwono – czarne;

Drzewa czerwono – czarne (r-b)

- Drzewo czerwono – czarne „r-b” (1):
  - Drzewo czerwono – czarne (r-b) jest kolejnym rozwinięciem drzewa BST;
  - Powstało w celu przyspieszenia operacji przez odpowiednią organizację węzłów w drzewie;
  - Dzięki zastosowanym metodom równoważenia pesymistyczna złożoność operacji wynosi $O(lg\ n)$;
  - Drzewo r-b powstaje przez rozszerzenie węzła drzewa BST o pole koloru {czerwony, czarny};

Drzewa czerwono – czarne (r-b)

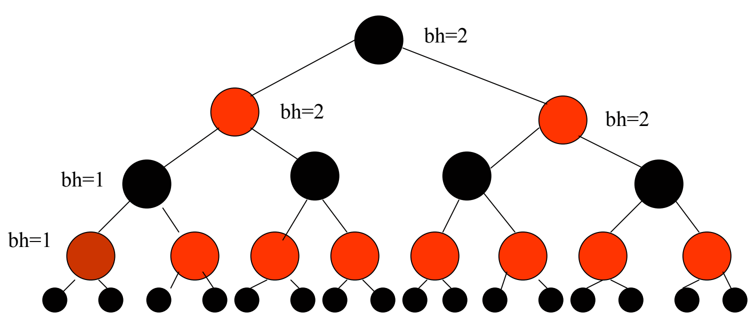
- Drzewo czerwono – czarne „r-b” (2):
  - W drzewie r-b:
    - Wszystkie wskazania NULL traktujemy jako wskazania na <u>zewnętrzne węzły</u> drzewa – <u>liście</u>;
    - Zwyczajne węzły drzewa r-b (zawierające klucze) nazywamy <u>wewnętrznymi węzłami</u> drzewa;
  - Przyjmuje się następujące założenia:
    - (1) Każdy węzeł ma kolor czerwony lub czarny;
    - (2) Korzeń ma kolor czarny;
    - (3) Każdy wskaźnik o wartości NULL ma kolor czarny;
    - (4) Jeżeli węzeł ma atrybut czerwony, to jego potomkowie są koloru czarnego;
    - (5) Dla każdego węzła każda prosta ścieżka od węzła do liścia zawiera jednakową liczbę węzłów czarnych;

Drzewa czerwono – czarne

- Drzewo czerwono – czarne (3):
  - Wysokość węzła: h(x) – największa liczba węzłów na drodze prostej z węzła x do liści;
  - Czarna wysokość węzła: bh(x) – liczba węzłów czarnych (z uwzględnieniem NULL) na drodze od tego węzła do liścia (z wykluczeniem tego węzła);
  - Czarna wysokość drzewa r-b – czarna wysokość korzenia danego drzewa;

Drzewa czerwono – czarne

- Drzewo czerwono – czarne (3):
  - Przykład drzewa:



Drzewa czerwono – czarne

- Twierdzenia dotyczące drzewa czerwono – czarnego (1):
  - Lemat 1
    - Każde poddrzewo r-b o korzeniu w dowolnym *x* posiada co najmniej $n = 2^{bh(x)} - 1$ wewnętrznych węzłów.
  - Dowód:
    - Przez indukcję względem wysokości węzła *x*:
      - Jeżeli *x* ma wysokość 0, to *x* musi być liściem (NULL) a poddrzewo o korzeniu w *x* zawiera $2^{bh(x)}-1=2^0-1=0$ węzłów;
      - Niech *x* będzie węzłem wewnętrznym o dodatniej wysokości i dwóch potomkach. Każdy ma czarną wysokość równą bh(x) albo bh(x)-1 (zależnie od koloru);
        Wysokość potomka jest mniejsza od wysokości węzła, więc każde z poddrzew o korzeniach w potomkach *x* ma ma co najmniej $2^{bh(x)-1}-1$ węzłów wewnętrznych;
      - Zatem drzewo o korzeniu *x* ma:
        $(2^{bh(x)-1}-1)+(2^{bh(x)-1}-1)+1 = 2^{bh(x)}-1$.

Krok indukcyjny

Drzewa czerwono – czarne

- Twierdzenia dotyczące drzewa czerwono – czarnego (2):
  - Lemat 2
    - Każdy węzeł *x* o wysokości *h(x)* ma czarną wysokość równą bh(x) ≥ h(x) / 2.
  - Dowód
    - Z założenia 4 wynika, że co najwyżej h(x) / 2 węzłów na drodze z węzła do liścia jest czerwona;
    - Zatem co najmniej h(x) / 2 węzłów jest czarna;
      Inaczej: czarna wysokość węzła wynosi co najmniej h(x)/2;

Drzewa czerwono – czarne

- Twierdzenia dotyczące drzewa czerwono – czarnego (3):
  - Lemat 3
    - Drzewo r-b o *n* węzłach wewnętrznych ma wysokość nie większą niż *2 lg(n+1).*
  - Dowód:
    - Niech h(x) będzie wysokością drzewa o korzeniu x.
    - Z lematu 2: $bh(x) \geq h(x)/2$
    - Z lematu 1: $n \geq 2^{bh(x)} - 1$
    - Zatem:
      - $n \geq 2^{h(x) / 2} - 1$
      - $n + 1 >= 2^{h(x) / 2}$
      - $\lg(n + 1) \geq h(x) / 2$
      - $2\lg(n + 1) \geq h(x)$.

Operacje na drzewie czerwono – czarnym

- ***Wyszukanie, minimum, maksimum, następnik, poprzednik:***
  - ponieważ drzewo tego typu jest drzewem BST, te operacje wyglądają tak, jak dla drzew BST;
  - złożoność operacji:
    - dla BST: *O(h)*;
    - dla r-b: *O(lg n)*;
- ***Wstawianie, usuwanie:***
  - te operacje ulegają zmianom;
  - dlaczego?

Operacje na drzewie czerwono – czarnym

- ***Wstawianie:***
  - Wstawienie analogiczne do operacji w drzewie BST;
  - Korekta: wyróżniamy dwa podejścia: „***Bottom Up***” oraz ***„Top-down***”;
  - Jaki kolor nadać wstawianemu węzłowi?
  - Czerwony (?):
    - Może zostać zaburzone założenie 4:
      - *Jeżeli węzeł ma atrybut czerwony, to jego potomkowie są koloru czarnego;*
  - Czarny (?):
    - Może zostać zaburzone założenie 5:
      - *Dla każdego węzła każda ścieżka od węzła do liści zawiera jednakową liczbę węzłów czarnych;*

Operacje na drzewie czerwono – czarnym

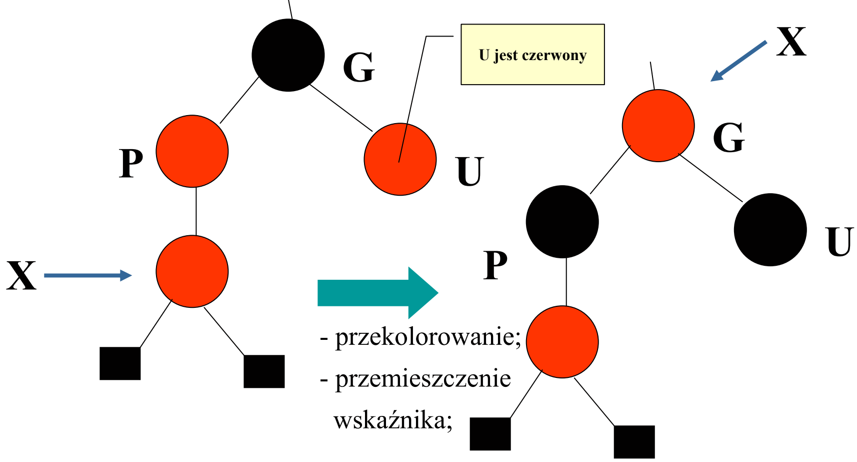
- ***Wstawianie „Bottom Up” (1):***
  - *Wstaw węzeł;*
  - *Pokoloruj węzeł na czerwono (X wskazuje na nowy węzeł);*
  - *Możliwe przypadki:*
    - *0: X to korzeń → koloruj na czarno;*
    - *1: Zarówno przodek (ojciec) jak i bezpośredni sąsiad przodka (wuj) są czerwone;*
    - *2 (zig-zag): Przodek jest czerwony, ale wuj jest czarny; Ponadto X i ojciec są synami po przeciwnych stronach;*
    - *3 (zig-zig): Przodek jest czerwony, ale wuj jest czarny; Ponadto X i ojciec są albo lewymi albo prawymi synami;*

Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up” – przypadek 1 (1)***
  - *(1): Zarówno przodek (ojciec) jak i bezpośredni sąsiad przodka (wuj) są czerwone*
  - *Korekta:*
    - *koloruj przodka i jego sąsiada (wuja) na czarno;*
    - *koloruj przodka przodka (dziadka) na czerwono;*
    - *wskaż X na przodka przodka (dziadka);*

Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up” –*** przypadek 1 (2)

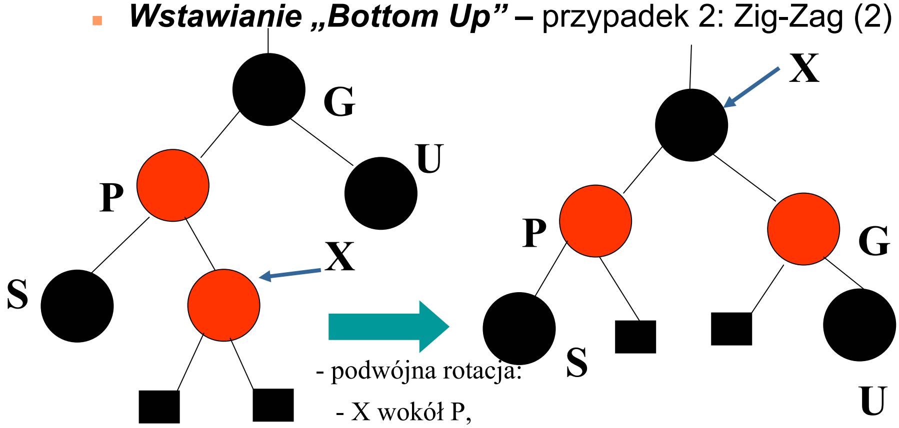


Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up”* –** przypadek 2: Zig-Zag (1)
  - *(2) Zig-Zag:*
    - *Przodek jest czerwony, ale wuj jest czarny;*
    - *Ponadto X i ojciec są synami po przeciwnych stronach;*
  - Korekta:
    - wykonaj rotację w lewo wokół ojca;
    - wykonaj rotację w prawo wokół dziadka;
    - koloruj dziadka na kolor czerwony;
    - koloruj X na czarno;

Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up”* –** przypadek 2: Zig-Zag (2)



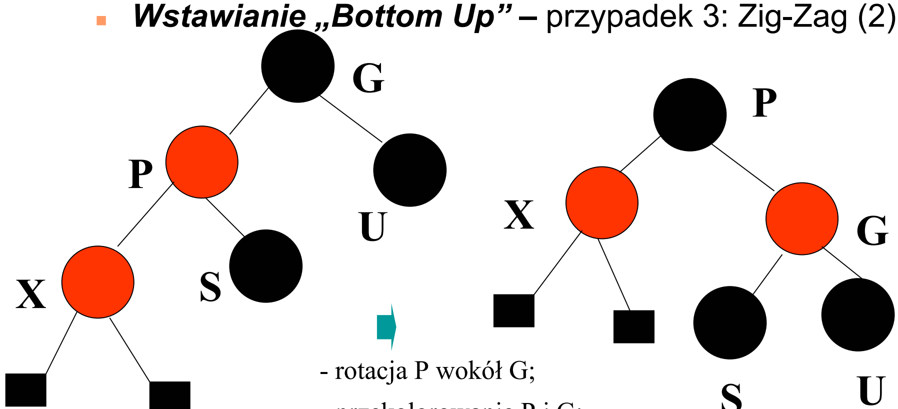
- podwójna rotacja:
  - X wokół P,
  - X wokół G;
- przekolorowanie G i X;

Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up” –*** przypadek 3: Zig-Zag (1)
  - *(3) Zig-Zag:*
    - *Przodek jest czerwony, ale wuj jest czarny;*
    - *Ponadto X i ojciec są albo lewymi albo prawymi synami*
  - Korekta:
    - wykonaj rotację w prawo wokół dziadka;
    - koloruj ojca i dziadka na kolor czarny;

Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up” –*** przypadek 3: Zig-Zag (2)

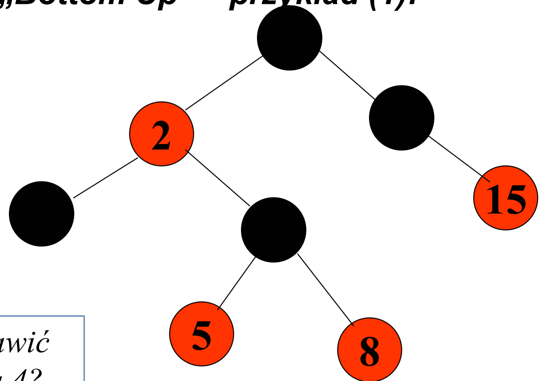


- rotacja P wokół G;
- przekolorowanie P i G;

→ *Ten przypadek zachodzi w drugim kroku przypadku 2;*

Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up” – przykład (1):***



Czy potrafisz wstawić węzeł z wartością 4?

Operacje na drzewie czerwono – czarnym

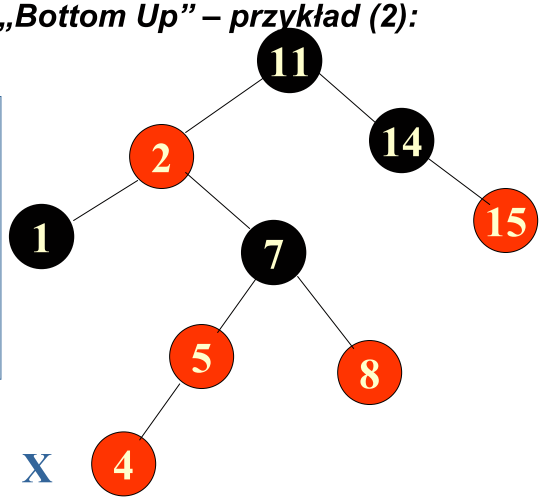
- ***Wstawianie „Bottom Up” – przykład (2):***

Po wstawieniu węzła 4:

- Węzeł x i przodek są koloru czerwonego;

*- Mamy **przypadek 1**;*

- Zatem należy przekolorować węzły i przesunąć wskaźnik x;



Operacje na drzewie czerwono – czarnym

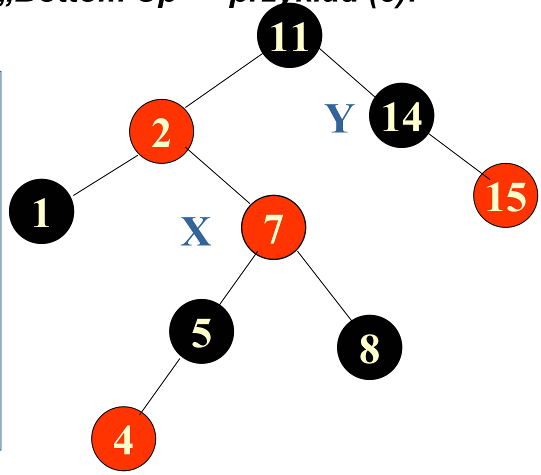
- ***Wstawianie „Bottom Up” – przykład (3):***

- Ponownie węzeł x i przodek są koloru czerwonego;

- Stryj y węzła x jest czarny a x jest prawym synem;

*- Mamy **przypadek 2**;*

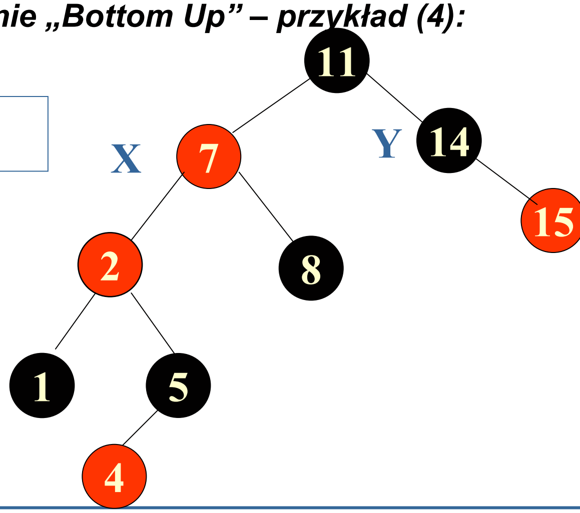
- Zatem należy dokonać podwójnej rotacji x i przekolorować węzły;



Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up” – przykład (4):***

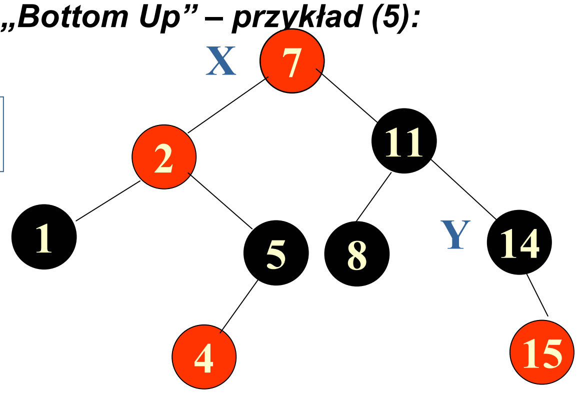
-Po wykonaniu lewej rotacji x względem 2;



Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up" – przykład (5):***

-Po wykonaniu prawej rotacji x względem 11;

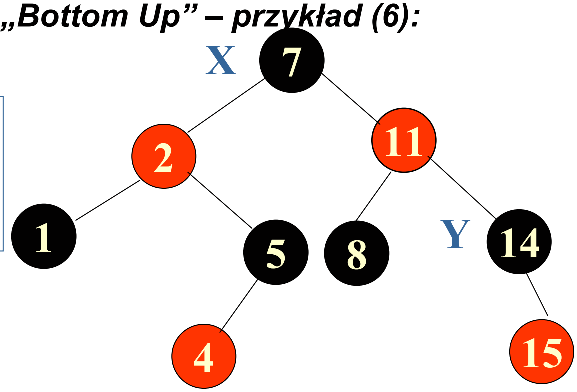


Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up" – przykład (6):***

-Po przekolorowaniu x i 11;

-Drzewo teraz jest poprawnym drzewem r-b;



Operacje na drzewie czerwono – czarnym

- ***Wstawianie „Bottom Up” – algorytm korekty:***

```
while color[p[z]] = RED
    do if p[z] = left[p[p[z]]]
          then y ← right[p[p[z]]]
               if color[y] = RED
                 then color[p[z]] ← BLACK
                      color[y] ← BLACK
                      color[p[p[z]]] ← RED
                      z ← p[p[z]]
                 else if z = right[p[z]]
                        then z ← p[z]
                             LEFT-ROTATE(T, z)
                      color[p[z]] ← BLACK
                      color[p[p[z]]] ← RED
                      RIGHT-ROTATE(T, p[p[z]])
          else (same as then clause
                      with "right" and "left" exchanged)
color[root[T]] ← BLACK
```

Przypadek (1) – lines 5–8

Przypadek (2) – lines 10–11

Przypadek (3) – lines 12–14

Operacje na drzewie czerwono – czarnym

- ***Usuwanie:***
  - Jakie zmiany wywołuje usunięcie węzła?
  - Czerwonego:
    - Czarne wysokości węzłów nie zmieniają się;
    - Ponadto usuwany węzeł nie mógł być korzeniem, bo był koloru czerwonego;
  - Czarnego:
    - Ścieżki z usuwanym węzłem mają o jeden węzeł czarny mniej: złamanie założeń 4 i 5;
    - Jeżeli usuwany węzeł był korzeniem, może zastąpić go potomek koloru czerwonego: złamanie zasady 2;

Operacje na drzewie czerwono – czarnym

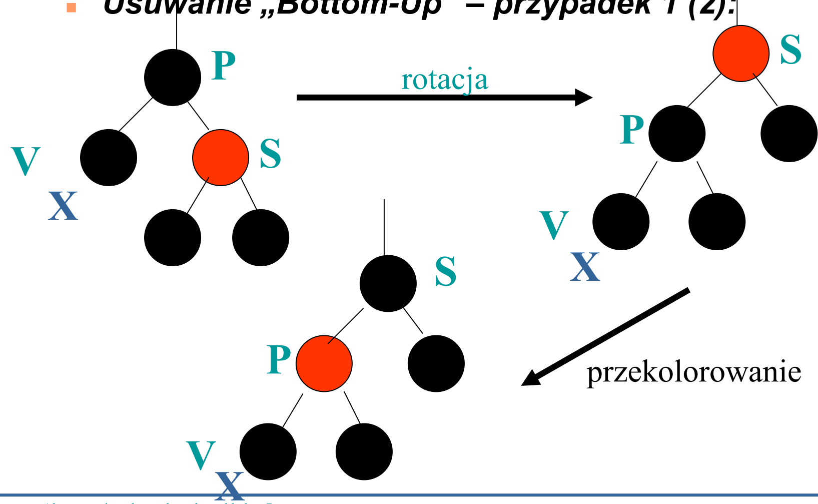
- ***Usuwanie „Bottom-Up”:***
  - Usuwanie analogiczne do operacji w drzewie BST;
  - Korekta przy usuwaniu węzła czarnego:
    - Założenia:
      - U – węzeł usuwany;
      - V – węzeł, który zastępuje U;
      - P – przodek węzła V;
      - S – sąsiad (brat) węzła V;
      - X wskazuje węzeł z nadmiarowym kolorem czarnym;
    - Przypadki:
      - (1) S jest koloru czerwonego;
      - (2) S jest koloru czarnego oraz ma dwóch czarnych potomków;
      - (3) S jest koloru czarnego a prawy potomek koloru czerwonego;
      - (4) S jest koloru czarnego, prawy potomek koloru czarnego a lewy koloru czerwonego;

Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up” – przypadek 1 (1):***
  - (1) S jest koloru czerwonego;
  - Korekta:
    - Rotacja S wokół P;
    - Przekolorowanie S i P;
  - Uwaga – to nie jest stan końcowy;
  - Wystąpi jeden z przypadków (2) – (4);

Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up" – przypadek 1 (2):***

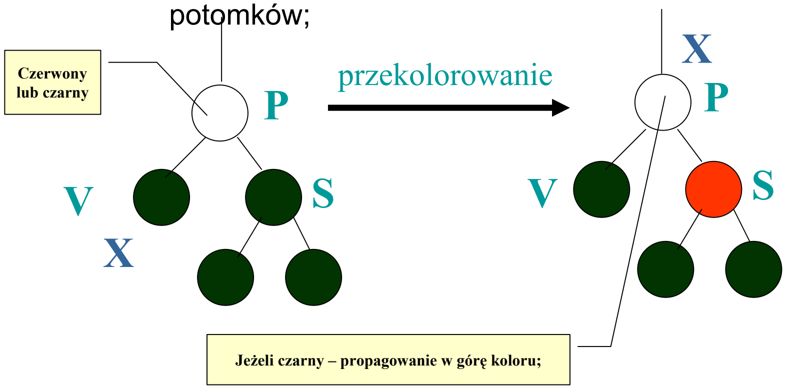


Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up" – przypadek 2 (1):***
  - (2) S jest koloru czarnego oraz ma dwóch czarnych potomków;
  - Korekta:
    - Przekolorowanie S na czerwono;
    - Jeżeli P jest koloru czerwonego – bez zmian;
    - Jeżeli P jest czarne – kolor musi być propagowany w drzewie w kierunku korzenia;
      - Korekta musi być rekurencyjnie powtarzana aż do wyprowadzenia koloru nadmiarowego z drzewa;

Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up" – przypadek 2 (2):***
  - (2) S jest koloru czarnego oraz ma dwóch czarnych potomków;

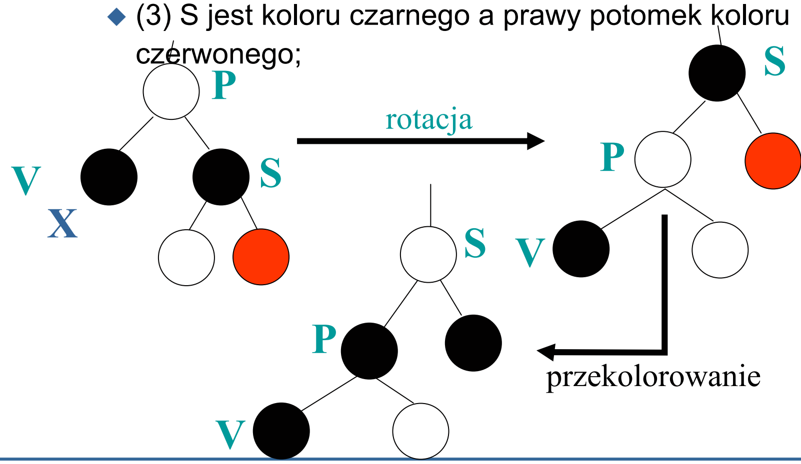


Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up" – przypadek 3 (1):***
  - (3) S jest koloru czarnego a prawy potomek koloru czerwonego (lewy – dowolnego);
  - Korekta:
    - Rotacja S wokół P;
    - Zamiana kolorów S i P;
    - Przekoloruj prawego potomka S na czarno;
  - Przypadek końcowy!

Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up” – przypadek 3 (2):***
  - (3) S jest koloru czarnego a prawy potomek koloru czerwonego;

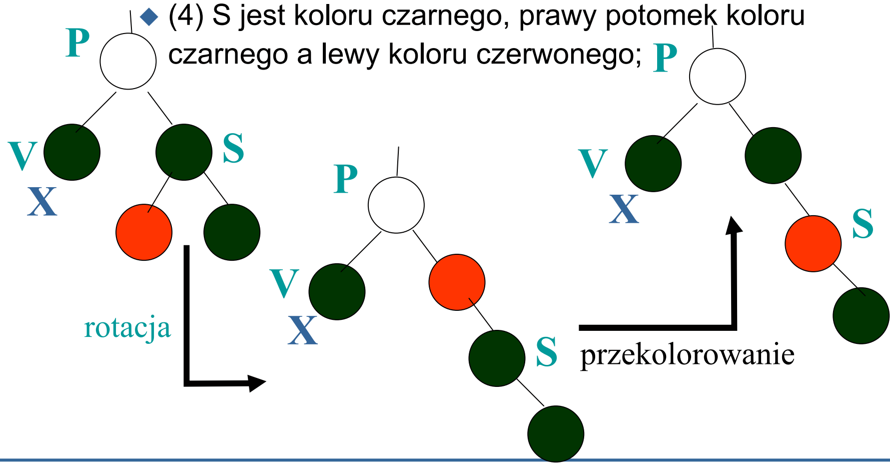


Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up” – przypadek 4 (1):***
  - (4) S jest koloru czarnego, prawy potomek koloru czarnego a lewy koloru czerwonego;
  - Korekta:
    - Rotacja lewego potomka S wokół S;
    - Zamiana kolorów S i lewego potomka S;
  - Teraz stan zgodny z przypadkiem 3;

Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up" – przypadek 4 (2):***
  - (4) S jest koloru czarnego, prawy potomek koloru czarnego a lewy koloru czerwonego;



Operacje na drzewie czerwono – czarnym

- ***Usuwanie „Bottom-Up" – algorytm korekty:***

```
while x ≠ root[T] and color[x] = BLACK
    do if x = left[p[x]]
        then w ← right[p[x]]
            if color[w] = RED
              then color[w] ← BLACK                              ▷ Case 1
                   color[p[x]] ← RED                              ▷ Case 1
                   LEFT-ROTATE(T, p[x])                           ▷ Case 1
                   w ← right[p[x]]                                ▷ Case 1
            if color[left[w]] = BLACK and color[right[w]] = BLACK
              then color[w] ← RED                                 ▷ Case 2
                   x ← p[x]                                       ▷ Case 2
              else if color[right[w]] = BLACK
                       then color[left[w]] ← BLACK                ▷ Case 3
                            color[w] ← RED                        ▷ Case 3
                            RIGHT-ROTATE(T, w)                    ▷ Case 3
                            w ← right[p[x]]                       ▷ Case 3
                   color[w] ← color[p[x]]                         ▷ Case 4
                   color[p[x]] ← BLACK                            ▷ Case 4
                   color[right[w]] ← BLACK                        ▷ Case 4
                   LEFT-ROTATE(T, p[x])                           ▷ Case 4
                   x ← root[T]                                    ▷ Case 4
        else (same as then clause with "right" and "left" exchanged)
color[x] ← BLACK
```

Przypadek (1)

Przypadek (2)

Przypadek (3)

Przypadek (1)

Dziękuję za uwagę