Drzewa binarne

Drzewo binarne to dowolny obiekt powstały zgodnie z regułami:

- $\perp$ jest drzewem binarnym
- Jeśli T_0 i T_1 są drzewami binarnymi to $T_0 \wedge T_1$ jest drzewem binarnym

Np. $(\perp \wedge \perp) \wedge (\perp \wedge (\perp \wedge \perp))$

Wielkość drzewa binarnego

- $|\perp| = 1$
- $|T_0+T_1| = |T_0| + |T_1| + 1$

Szerokość drzewa binarnego

- $w(\perp) = 1$
- $w(T_0 \wedge T_1) = w(T_0) + w(T_1)$

Wysokość drzewa binarnego

- $h(\bot) = 0$
- $h(T_0 \wedge T_1) = \max(h(T_0), h(T_1)) + 1$

Własności

- $h(T_i) < h(T)$
- $w(T_i) < w(T)$
- $|T_i| < |T|$
- $h(T) \leq w(T) \leq |T|$
- $|T| = 2*w(T) - 1$ (udowodnić indukcyjnie)

Przechodzenie drzewa

- t, T_L, T_R - *pre-order*, przejście wzdłużne
- T_L, t, T_R - *in-order*, przejście poprzeczne (przejście grafu w głąb)
- T_L, T_R, t - *post-order*, przejście wsteczne

Drzewo przeszukiwań (*search tree*)

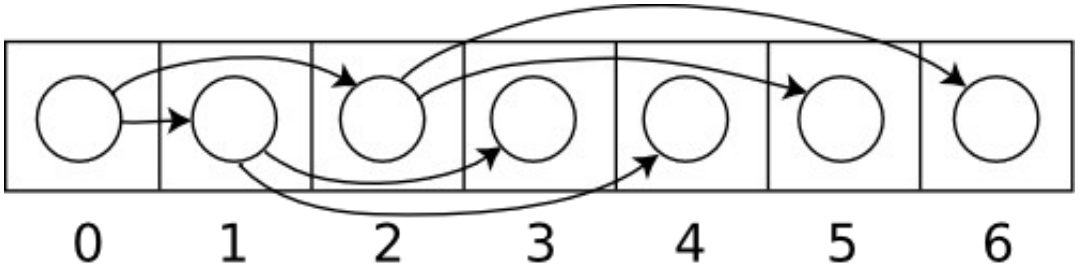
- Dla każdego węzła t, wszystkie klucze w lewym poddrzewie są mniejsze od klucza w t, a wszystkie klucze w prawym poddrzewie są większe od klucza w t.
- Przejście *in-order* po drzewie przeszukiwań daje nam posortowany ciąg kluczy.

Drzewo zrównoważone

- Jest to drzewo, w którym głębokość dowolnych dwóch liści różni się co najwyżej o jeden.
- Wyszukanie elementu w zrównoważonym binarnym drzewie przeszukiwań wymaga $O(\log_2 n)$ operacji (w drzewie o wysokości n możemy zmieścić 2^n-1 węzłów).

Reprezentacja drzewa

W tablicy – wybrany węzeł ma indeks i. Korzeń lewego poddrzewa ma indeks 2i+1, a prawego 2i+2:



Powyższa metoda jest dobra dla drzew zrównoważonych (mocno „wypełnionych”).

Możemy też drzewa reprezentować za pomocą struktur zawierających wskaźniki do potomków.

Źródło obrazka: wikipedia

Przeszukiwanie drzewa (bez wartownika)

```
PROCEDURE locate(x: INTEGER; t: Node): Node;
BEGIN
   WHILE (t # NIL) & (t.key # x) DO
      IF t.key < x THEN
         t := t.right
      ELSE
         t := t.left
      END
   END ;
   RETURN t
END locate
```

Źródło: N.Wirth

Przeszukiwanie drzewa (z wartownikiem)

```
PROCEDURE locate(x: INTEGER; t: Ptr): Ptr;
BEGIN
   s.key := x; (*sentinel*)
   WHILE t.key # x DO
      IF t.key < x THEN
         t := t.right
      ELSE
         t := t.left
      END
   END;
   RETURN t
END locate
```

Źródło: N.Wirth

Dodawanie elementu do drzewa (*tree search with insertion*)

```
PROCEDURE search(x: INTEGER; VAR p: Node);
BEGIN
   IF p = NIL THEN (*x not in tree; insert*)
      NEW(p);
      p.key := x;
      p.count := 1;
      p.left := NIL;
      p.right := NIL
   ELSIF x < p.key THEN
      search(x, p.left)
   ELSIF x > p.key THEN
      search(x, p.right)
   ELSE
      INC(p.count)
   END
END search;
```

Źródło: N.Wirth

Usuwanie elementu z drzewa

```
PROCEDURE delete(x: INTEGER; VAR p: Node);
VAR q: Node;
   PROCEDURE del (VAR r: Node);
   BEGIN
      IF r.right # NIL THEN del(r.right)
      ELSE q.key := r.key; q.count := r.count;
         q := r; r := r.left
      END
   END del;
   BEGIN (*delete*)
      IF p = NIL THEN (*word is not in tree*)
      ELSIF x < p.key THEN delete(x, p.left)
      ELSIF x > p.key THEN delete(x, p.right)
      ELSE (*delete p^*) q := p;
      IF q.right = NIL THEN p := q.left
      ELSIF q.left = NIL THEN p := q.right
      ELSE del(q.left)
      END
   END
END delete
```

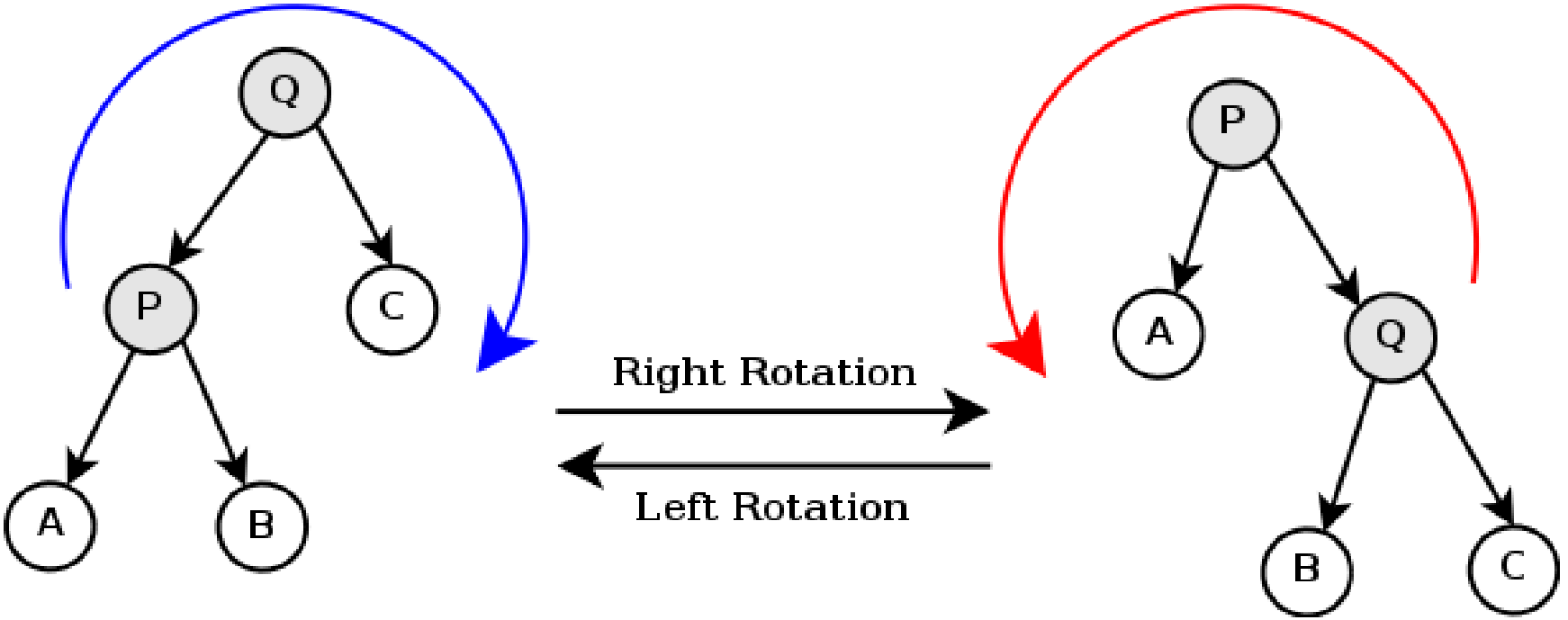
Źródło: N.Wirth

Równoważenie drzewa

- Za pomocą pomocniczej tablicy.
- Przechodzimy drzewo w kolejności *in-order* – uzyskujemy posortowaną tablicę.
- Z tablicy tworzymy nowe drzewo, środkowy element zostaje korzeniem.
  Z elementów na lewo od niego tworzymy rekurencyjnie lewe poddrzewo, z elementów na prawo – prawe poddrzewo.

Algorytm DSW

- Day/Stout/Warren 1986
- Złożoność czasowa O(n)
- Algorytm działa „w miejscu” (*in situ*)



Źródło obrazka: wikipedia

Algorytm DSW

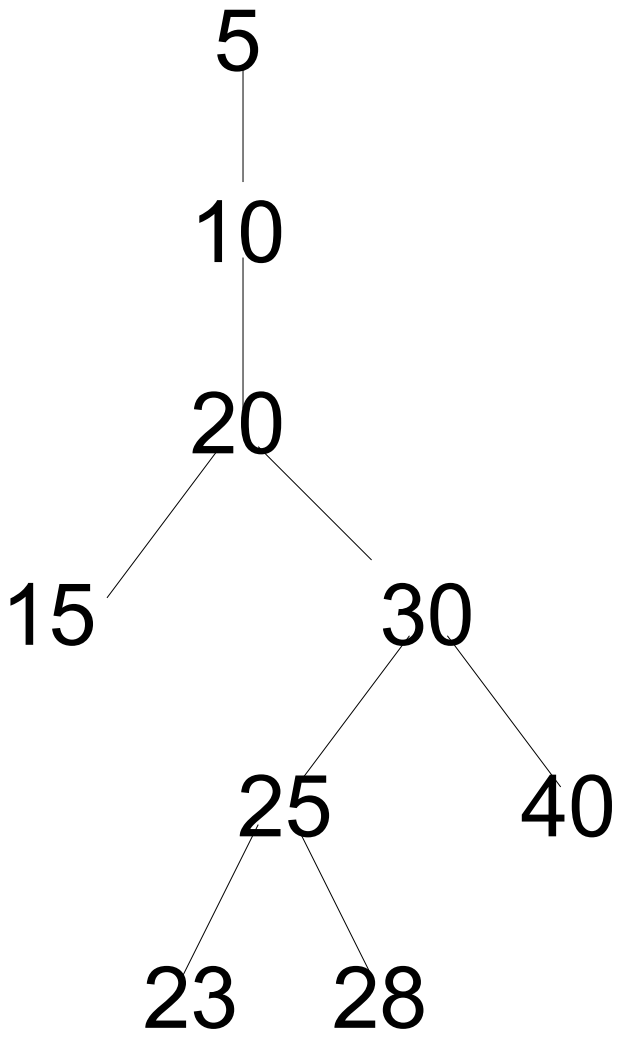
- Najpierw wykonujemy obroty w prawo dla korzenia, a później jego wszystkich potomków. Uzyskujemy w ten sposób listę.
- Następnie wykonujemy obroty w lewo dla co drugiego węzła znajdującego się w prawym poddrzewie. Ilość obrotów jest określona:
  - W pierwszym przebiegu leaf_count = $n+1-2^{\lfloor \lg(n+1) \rfloor}$
  - W drugim przebiegu $(n\text{-}leaf\_count)$ div 2
  - W kolejnych przebiegach dzielimy przez 2

Algorytm DSW

```
begin
   leaf-count := size + 1 - 2^⌊lg(size+1)⌋;
   compression (root, leaf-count);
   size := size - leaf-count;
   while size > 1 do begin
      compression (root, size div 2);
      size := size div 2;
   end;
end;
```

Źródło: Q.F.Stout, B.L.Warren „Communications of the ACM”, september 1986, vol.29 n.9

Algorytm DSW - przykład



Czy opłaca się równoważenie drzewa?

- Wyszukanie w najgorszym przypadku (lista) wymaga O(n) operacji.
- Jednak szanse na powstanie listy są małe (możliwe jest n! kolejności podania elementów).
- Dla dużych n średnia długość ścieżki wyszukiwania to $a_n = 2*(\ln n + g - 1)$
- g – stała Eulera = 0.577...
- W drzewie zrównoważonym: $a_n' = \log n - 1$
- $\lim(a_n/a_n') = 2*\ln(n)/\log(n) = 2*\ln(2) = 1.386...$
- Średnio zyskamy więc ok. 39%

Drzewa AVL

- Adelson-Velskii & Landis 1962
- Drzewo jest zrównoważone wtedy i tylko wtedy, gdy dla każdego węzła różnica wysokości jego dwóch poddrzew wynosi co najwyżej jeden.
- Nie musimy równoważyć całego drzewa po każdej operacji dodania/usuwania (jak w DSW).
- Drzewo AVL jest co najwyżej o 45% wyższe od drzewa idealnie zrównoważonego.
- Drzewo AVL o najmniejszej dozwolonej ilości węzłów tworzy drzewo Fibonacciego.

Drzewa Fibonacciego

- Puste drzewo to drzewo Fibonacciego o wysokości 0.
- Pojedynczy węzeł to drzewo Fibonacciego o wysokości 1.
- Jeśli T_{h-1} i T_{h-2} są drzewami Fibonacciego o wysokościach odpowiednio h-1 i h-2 to $T_h = T_{h-1} \wedge T_{h-2}$ jest drzewem Fibonacciego.
- Ilość węzłów: $N_0 = 0 \qquad N_1 = 1$

$$N_h = N_{h-1} + 1 + N_{h-2}$$

Drzewa AVL

- Dla każdego węzła zapamiętujemy współczynnik wyważenia równy różnicy wysokości prawego i lewego poddrzewa. Może on wynosić 0, +1 lub -1.
- Wstawianie do drzewa AVL może odbywać się tak, jak gdyby było ono zwyczajnym drzewem poszukiwań binarnych, następnie zaś, odtwarzając kroki w kierunku korzenia, wykonywane są aktualizacje wyważeń węzłów.

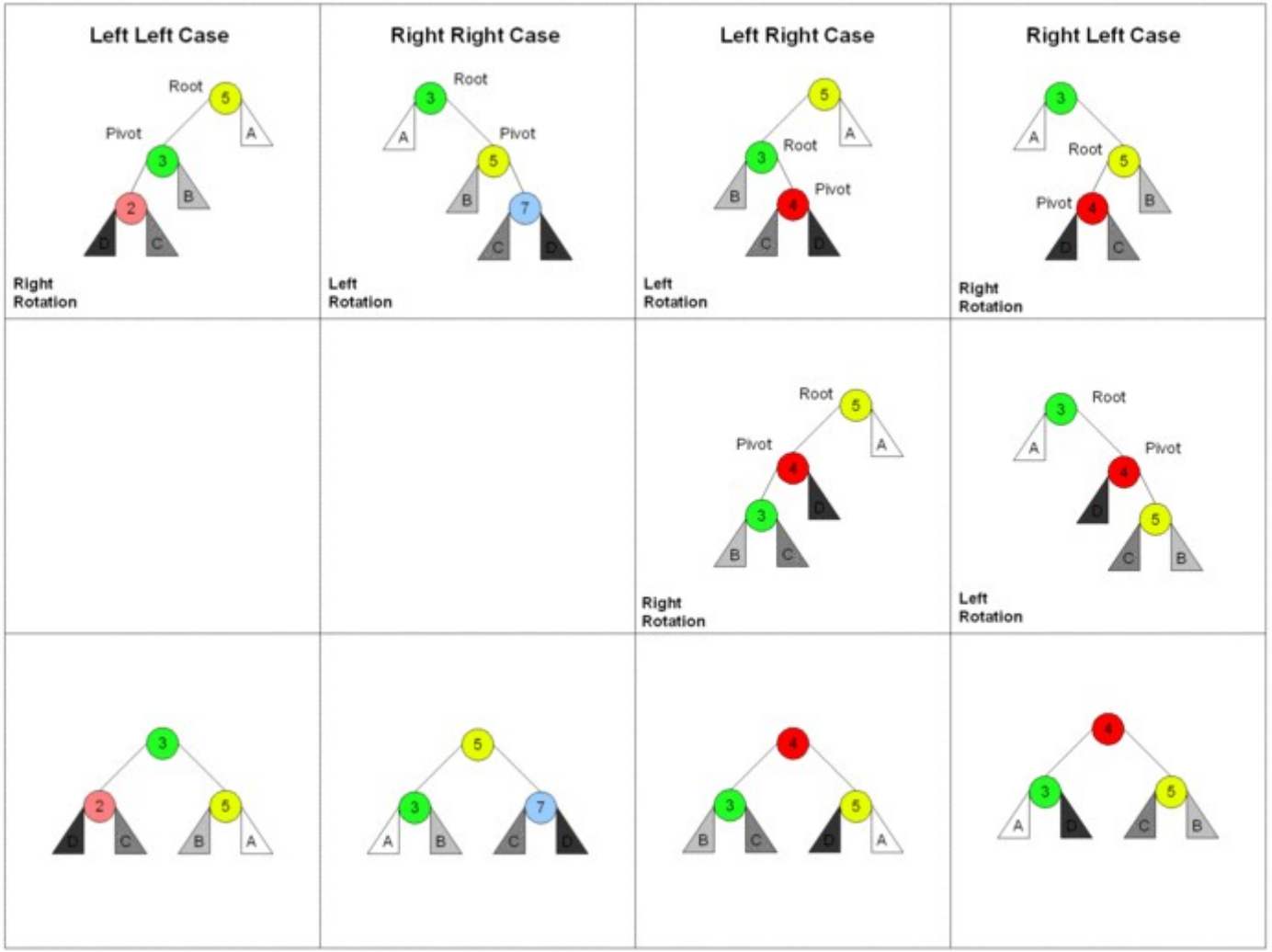
Drzewa AVL

- Jeśli współczynnik został zmieniony na 2 lub -2, to drzewo straciło własność AVL. Aby ją przywrócić, potrzebna jest rotacja. Rotacja drzewa będzie zawsze zostawiać poddrzewo zrównoważone, nie ma potrzeby wykonywania dalszych aktualizacji.
- Z praktyki wynika, że drzewo AVL jest średnio o 25% wyższe niż idealnie zrównoważone, a operację równoważenia trzeba wykonać średnio co drugą operację wstawienia.

Drzewa AVL

There are 4 cases in all, choosing which one is made by seeing the direction of the first 2 nodes from the unbalanced node to the newly inserted node and matching them to the top most row.

Root is the initial parent before a rotation and **Pivot** is the child to take the root's place.



Źródło obrazka: wikipedia

Drzewa AVL

- Złożoność wyszukiwania+dodania O(log n) – dla DSW była O(n).
- Aplet demonstrujący działanie drzew AVL: http://webpages.ull.es/users/jriera/Docencia/AVL/AVL%20tree%20applet.htm

Usuwanie z drzewa AVL

- Jeśli usuwany węzeł jest liściem, zostaje usunięty.
- Jeśli nie jest liściem, musi zostać zastąpiony największym elementem z jego lewego poddrzewa lub najmniejszym z jego prawego poddrzewa. Wyszukany największy lub najmniejszy element ma co najwyżej jedno dziecko.
- Po usunięciu węzła odtwarzana jest ścieżka do korzenia, zaś współczynniki wyważenia są aktualizowane.

Usuwanie z drzewa AVL

- Odtwarzanie może być zatrzymane, jeśli współczynnik wyważenia zostaje zmieniony na -1 lub 1, oznacza to bowiem, iż wysokość poddrzewa pozostaje niezmieniona.
- Zmiana współczynnika wyważenia na 0 oznacza zmniejszenie wysokości poddrzewa, aktualizowanie współczynników musi być kontynuowane.
- Jeśli współczynnik zostanie zmieniony na -2 lub 2, to wykonywana jest rotacja w celu przywrócenia struktury AVL.

Usuwanie z drzewa AVL

- W przeciwieństwie do operacji wstawiania, wystąpienie rotacji nie musi oznaczać, iż drzewo zostało wyważone. W pesymistycznym przypadku rotacje będą wykonywane aż do wierzchołka drzewa.
- Z praktyki wynika, że operację równoważenia trzeba wykonać średnio co piątą operację usuwania.