ALGORYTMY
I STRUKTURY DANYCH

Temat 6:

Drzewa BST, AVL

Wykładowca: dr inż. Zbigniew TARAPATA

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

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

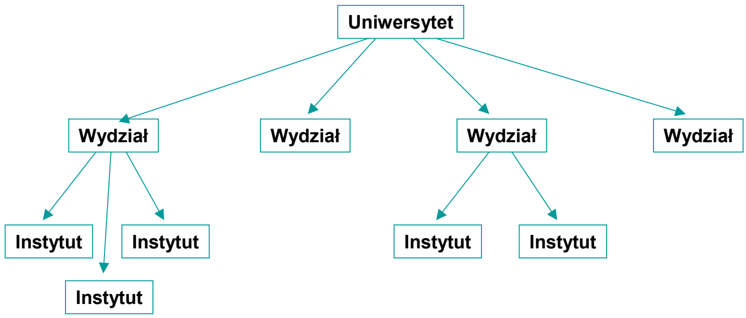
Współautorami wykładu są: A.Najgebauer, D.Pierzchała

Wykład : Algorytmy przetwarzania struktur drzewiastych

- drzewa poszukiwań binarnych;
- podstawowe operacje na drzewach BST;
- drzewa AVL;
- algorytmy rotacji;

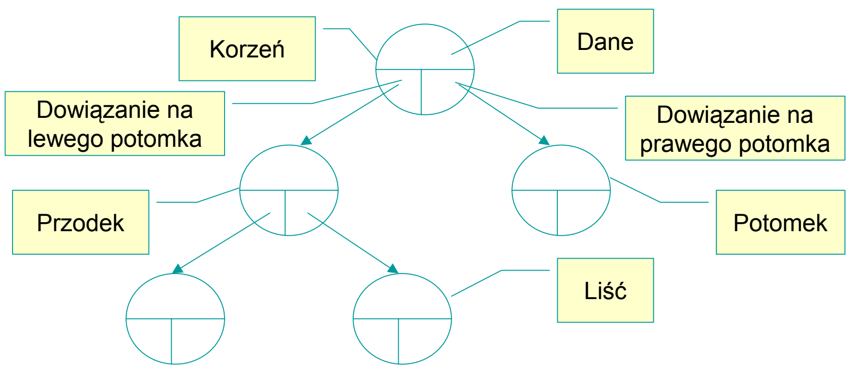
Przypomnienie – struktura drzewiasta

- Drzewiastą strukturą danych nazywamy strukturę danych ***S=(D, R, e)***, w której relacja porządkująca ***N*** opisuje kolejne, rekurencyjne powiązania pomiędzy danymi elementarnymi drzewa, tworzącymi kolejne „poddrzewa”.
- Przykład struktury drzewiastej



Element struktury drzewiastej

- Element drzewa zawiera:
  - Dane elementarne,
  - Realizację relacji następstwa – dowiązania do następników;



Drzewa poszukiwań binarnych

- Wysokość węzła *x*:
  - Największa liczba węzłów na drodze prostej z węzła *x* do liści (bez węzła x);
  - Inaczej jest to liczba krawędzi na tej drodze;
- Wysokość drzewa:
  - Wysokość korzenia;
  - *Wpływa na złożoność operacji na drzewie*;
- Głębokość węzła *x*:
  - Największa liczba węzłów na drodze prostej z korzenia do węzła *x* (bez korzenia);
  - Inaczej jest to liczba krawędzi na tej drodze;
- Głębokość drzewa:
  - Głębokość najodleglejszego liścia;

Drzewa poszukiwań binarnych

- Drzewo <u>binarne</u>:
  - Każdy węzeł ma co najwyżej dwóch potomków;
- Zupełne drzewo <u>binarne</u> (dwójkowe):
  - Każdy węzeł, z wyjątkiem liści, ma dokładnie dwóch potomków niepustych;
- Drzewo poszukiwań binarnych (<u>BST</u>):
  - Dla każdego węzła (nie liścia) wszystkie wartości przechowywane w lewym poddrzewie są mniejsze od jego wartości oraz przeciwnie dla drzewa prawego;
- Drzewo <u>binarne zrównoważone</u>:
  - Dla każdego węzła różnica wysokości obu poddrzew wynosi co najwyżej 1;
- Drzewo <u>doskonale zrównoważone</u>:
  - Wszystkie liście znajdują się na jednym poziomie;

Drzewa poszukiwań binarnych

- Ograniczenia w drzewie poszukiwań binarnych (BST):
  - Podstawowe operacje realizują się szybko, dzięki symetrycznemu porządkowi;
  - Jeśli jednak będziemy do drzewa wstawiać elementy np. posortowane, drzewo rozrośnie się w jedną ze stron (może w skrajnym przypadku być zdegenerowane do listy powiązanej);
  - Wtedy złożoność przeszukiwania (a tym samym wszystkich innych operacji) jest liniowa: *O(n)*;
  - Aby równoważyć drzewa BST wymyślono ich różne odmiany, np. drzewa ***AVL, czerwono-czarne, splay***;
  - Dzięki zrównoważeniu nie stracimy podstawowej zalety struktury drzewiastej: *złożoności obliczeniowej mniejszej niż liniowa*;

Drzewa poszukiwań binarnych

- Lemat 1
  - Liczba węzłów w drzewie doskonale zrównoważonym wynosi: 2^{h+1}-1;
- Dowód (przez indukcję)
  - Przyjmijmy, że dowolny poziom mają numer *l*, l=1..h;
  - Liczba wszystkich węzłów wynosi:

$$\sum_{l=0}^{h} 2^l = 2^{h+1} - 1$$

  - Przez indukcję:

dla *0:* $$\sum_{l=0}^{0} 2^l = 2^0 = 1 \Leftrightarrow 2^{0+1} - 1 = 2 - 1 = 1$$

a dla *H+1:* $$\sum_{l=0}^{H+1} 2^l = \sum_{l=0}^{H} 2^l + 2^{H+1} = 2^{H+1} - 1 + 2^{H+1} = 2^{H+2} - 1$$

Drzewa AVL

- Drzewo AVL (1962 – Adelson-Velskii, Landis)
  - Drzewo AVL jest rozwinięciem drzewa BST;
  - Dla każdego wierzchołka w drzewie AVL wysokości dwóch poddrzew różnią się o co najwyżej 1 poziom;
  - Węzeł oprócz pól danych, lewego i prawego dowiązania ma też pole opisujące różnicę wysokości lewego i prawego poddrzewa;
    - z definicji wynika, że to pole może mieć wartość z podzbioru {-1, 0, 1};

Operacje na drzewie AVL

- ***Wyszukanie:***
  - drzewo AVL jest też drzewem BST, zatem operacja wygląda tak jak dla drzew BST;
- ***Wstawianie:***
  - polega na wyszukaniu miejsca w drzewie a potem wstawieniu elementu (jak w BST);
  - ponieważ podczas operacji struktura drzewa zmienia się i może nie zostać zachowany warunek AVL (o różnicy w wysokości poddrzew), trzeba tę strukturę przywrócić;

Operacje na drzewie AVL

- ***Usuwanie:***
  - polega na wyszukaniu elementu w drzewie a potem jego usunięciu (patrz BST)
  - podczas operacji należy utrzymać zrównoważoną strukturę drzewa (j.w.);
- ***Rotacja:***
  - zmiana konfiguracji węzłów;
  - celem jest przywrócenie struktury drzewa AVL;
  - wyróżniamy rotacje:
    - lewe i prawe,
    - pojedyncze i podwójne;

Operacje na drzewie AVL

- ***Obliczanie wag wierzchołków:***
  - Dla każdego wierzchołka drzewa:

$$w(x) = h(LD) - h(PD),$$

gdzie LD i PD są odpowiednio lewym i prawym poddrzewem drzewa o korzeniu w x;

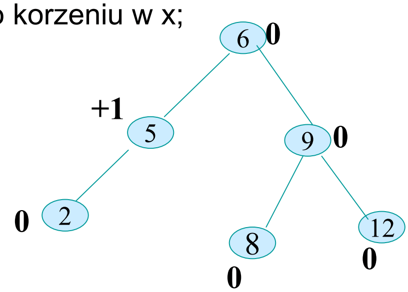
Pamiętaj!

Drzewo BST jest drzewem AVL

$\Leftrightarrow$

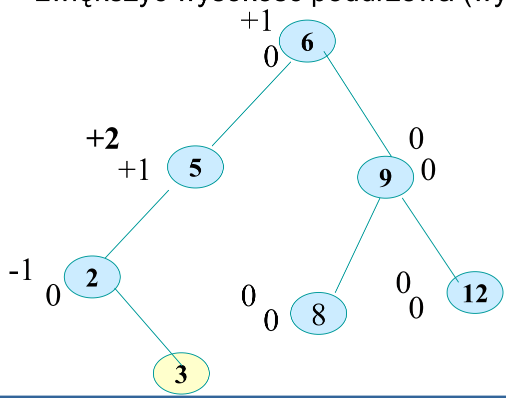
dla każdego wierzchołka *w*:

$w(x) \in \{-1, 0, +1\}$



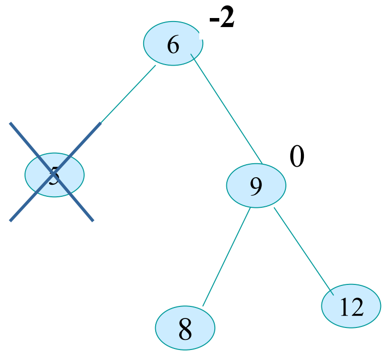
Operacje na drzewie AVL

- ***Wstawianie* – niespełniony warunek AVL*:***
  - Wstawienie nowego elementu do drzewa BST może zwiększyć wysokość poddrzewa (wymagana rotacja);



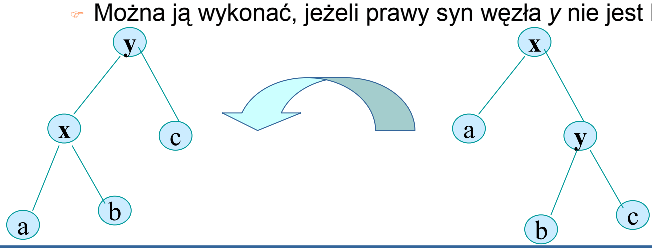
Operacje na drzewie AVL

- ***Usuwanie* – niespełniony warunek AVL*:***
  - Usunięcie elementu z drzewa BST może zmniejszyć wysokość poddrzewa (wymagana rotacja)!



Operacje na drzewie AVL

- ***Rotacja:***
  - Lewa rotacja (lub „w lewo”):
    - Polega na obrocie wokół wyróżnionego węzła (*x*) przeciwnie do ruchu wskazówek zegara;
    - W wyniku rotacji węzeł *y* staje się nowym korzeniem poddrzewa, węzeł *x* zostaje jego lewym synem a lewy syn węzła y zostaje prawym synem węzła *x*;
    - Można ją wykonać, jeżeli prawy syn węzła *y* nie jest NULL;

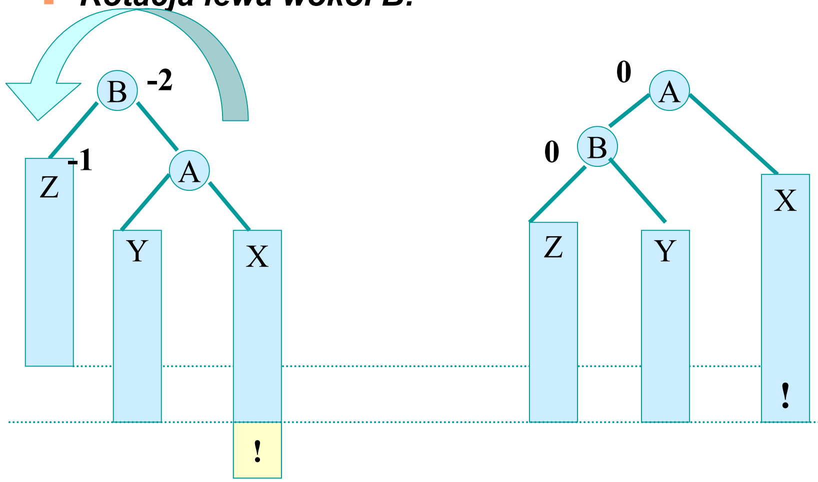


Operacje na drzewie AVL

- ***Rotacja:***
  - Prawa rotacja (lub „w prawo”):
    - Działa <u>symetrycznie</u> do lewej rotacji;
  - Rotację <u>rozpoczynamy</u> od węzła z wagą równą ± 2;
    - Kierunek zależy od znaku;
  - Przy <u>wstawianiu</u> elementu do drzewa AVL musimy wykonać co najwyżej 1 rotację;
  - Przy <u>usuwaniu</u> elementu z AVL może się zdarzyć, że będziemy musieli wykonać tyle rotacji ile jest poziomów w drzewie;

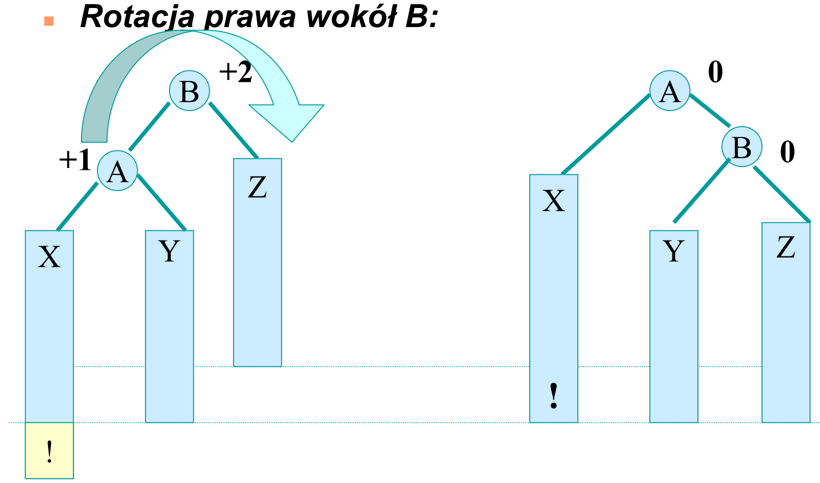
Operacje na drzewie AVL

- ***Rotacja lewa wokół B:***



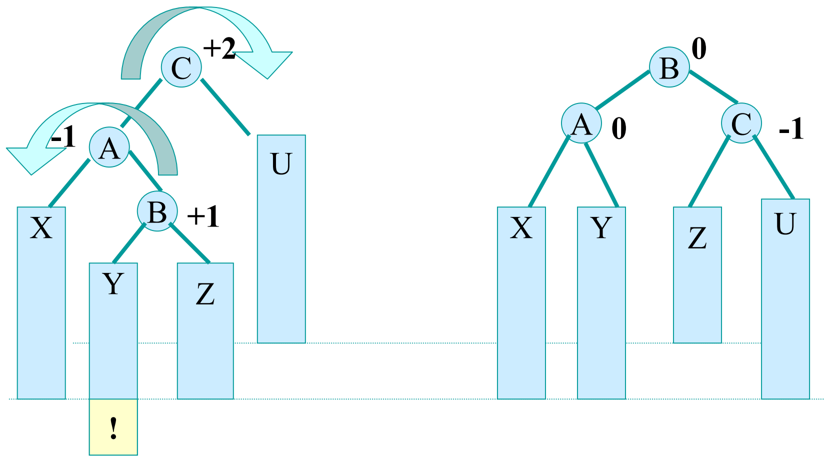
Operacje na drzewie AVL

- ***Rotacja prawa wokół B:***



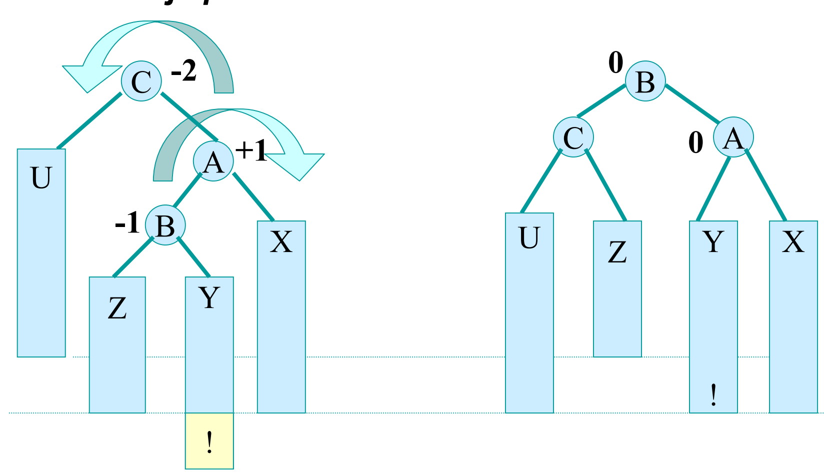
Operacje na drzewie AVL

- ***Rotacja lewa wokół A i prawa wokół C:***



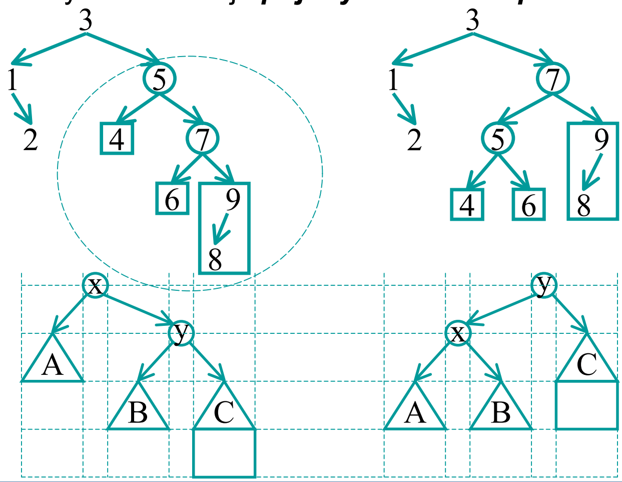
Operacje na drzewie AVL

- ***Rotacja prawa wokół A i lewa wokół C:***



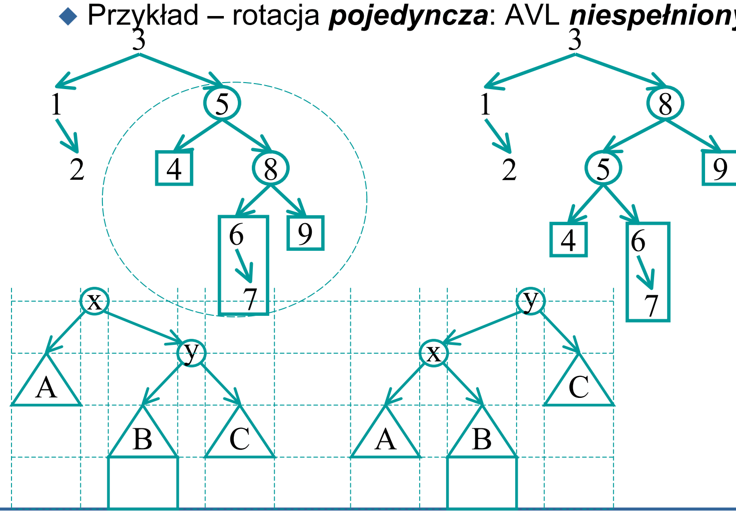
Operacje na drzewie AVL

- Przykład – rotacja ***pojedyncza***: AVL ***spełniony***



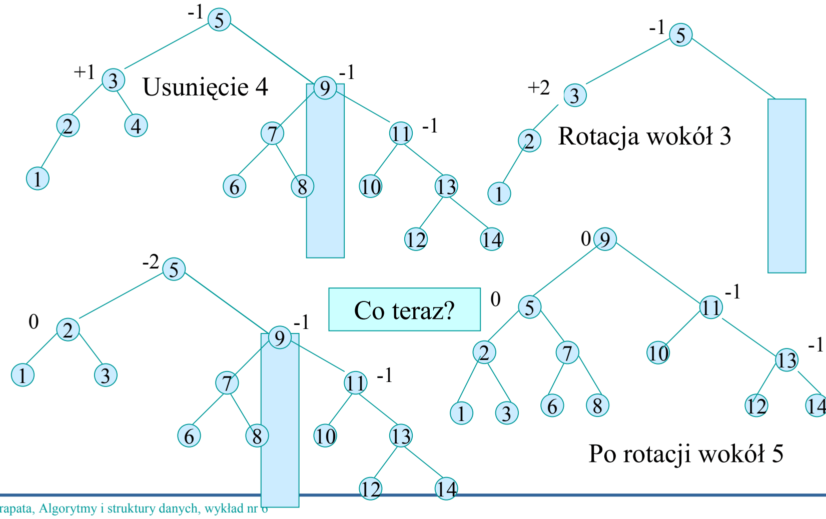
Operacje na drzewie AVL

- Przykład – rotacja ***pojedyncza***: AVL ***niespełniony***



Operacje na drzewie AVL

- Jeszcze jeden przykład – rotacja ***podwójna***: AVL ***spełniony***



Operacje na drzewie AVL

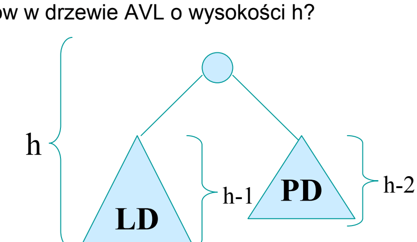
- ***Koszt*** operacji wstawiania i usuwania w AVL:
  - Rotacje działają w <u>czasie O(1)</u>;
  - Zmieniają tylko wartości wskaźników a pozostałe pola węzłów są bez zmian;
  - Dla drzewa o ‘n’ wierzchołkach:
    - minimalna liczba wierzchołków w drzewie AVL o wysokości h?

$N_0=1$

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

Można udowodnić przez indukcję, że $N_h \geq 2^{h/2}$

Stąd $h \leq 2 \lg N_h$



  - Liczba rotacji (koszt operacji) wynosi co najwyżej ‘lg n’;

Dziękuję za uwagę