Drzewa

Drzewa

- Drzewo (ang. tree) – zbiór węzłów powiązanych wskaźnikami, spójny i bez cykli.
- Drzewo posiada wyróżniony węzeł początkowy nazywany korzeniem (ang. root).
- Drzewo ukorzenione jest strukturą hierarchiczną.
- Korzeń węzła znajduje się na poziomie 0; numer poziomu danego węzła w drzewie jest wyznaczony odległością krawędziową od korzenia.
- Liściem (ang. leaf) w drzewie jest węzeł bez żadnego następnika.
- Węzeł wewnętrzny posiada co najmniej jednego następnika.
- Każdy węzeł oprócz korzenia posiada jednego poprzednika.

Drzewa

- Wysokość drzewa – długość najdłuższej ścieżki od korzenia do liścia (liczba węzłów) – inaczej liczba poziomów na których zapisane jest drzewo.
- Głębokość węzła – długość ścieżki od korzenia do tego węzła (liczba węzłów) – inaczej numer poziomu, na którym znajduje się węzeł.
- Uwaga: poziomy w drzewie numerujemy od 0; korzeń znajduje się na poziomie 0.

Drzewo

- Drzewo uporządkowane ma ponumerowanych (oznaczonych) następników.
- Drzewo uporządkowane składa się z węzłów, które zawierają następujące pola:
  - info – wartość pamiętana w węźle,
  - parent – wskaźnik na poprzednika,
  - left-child – wskaźnik na lewego syna (lista następników),
  - sibling – wskaźnik na prawego brata.
- Każdy węzeł w drzewie przechowuje jedną wartość.

Drzewo binarne

- Drzewo binarne (ang. binary tree) to drzewo ukorzenione, w którym każdy wierzchołek posiada dwóch rozróżnialnych synów – lewego i prawego.
- Węzeł drzewa binarnego zawiera pole z kluczem oraz wskaźnik na lewe poddrzewo (zakorzenione w jego lewym synu) i wskaźnik na prawe poddrzewo (zakorzenione w jego prawym synu). W drzewie binarnym poruszamy się tylko od korzenia w kierunku jakiegoś liścia.
- Jeśli w węźle drzewa binarnego znajduje się wskaźnik na ojca, to w drzewie takim można się przemieszczać w obu kierunkach – od korzenia w kierunku jakiegoś liścia oraz w drugą stronę w kierunku korzenia. O drzewie takim mówimy, że jest dwukierunkowe.
- Wysokość binarnego drzewa n-elementowego jest nie mniejsza niż log(n) i nie większa niż n.

Szczególne drzewa binarne

- Drzewo regularne to drzewo binarne, w którym każdy węzeł wewnętrzny na dwóch synów.
- Drzewo pełne to drzewo regularne, w którym wszystkie liście są na tym samym poziomie.
- Drzewo pełne o wysokości h ma 2^h-1 węzłów.
- Drzewo zupełne to drzewo pełne, z którego usunięto część liści z prawej strony.
- Drzewo zupełne o wysokości h ma od 2^{h-1} do 2^h-1 węzłów.
- Drzewo z wartownikiem – każdy wskaźnik pusty jest ustawiany na wartownika, a w wartowniku mamy wskaźnik do korzenia.

Drzewo BST

- Drzewo binarnych poszukiwań (ang. binary search tree), w skrócie BST, to drzewo binarne, w którym przechowujemy elementy z pewnego uniwersum z porządkiem liniowym i w drzewie tym jest zachowany porządek symetryczny.
- W drzewie binarnym jest zachowany porządek symetryczny, gdy elementy mniejsze od korzenia znajdują się w lewym poddrzewie, elementy większe od korzenia w prawym poddrzewie oraz w lewym i w prawym poddrzewie też jest zachowany porządek symetryczny.

Drzewo BST

- Wyszukiwanie w drzewie BST – działa podobnie do poszukiwania binarnego

```
serach (węzeł *w, x) -> boolean
{
	if (w == null) return false;
	if (x < w.info) return search(w.left, x);
	else if (w.info < x) return search(w.right, x);
	else return true;
}
```

Drzewo BST

- Element minimalny znajduje się najbardziej po lewej stronie w drzewie BST – od korzenia poruszamy się ciągle w lewo.
- Element maksymalny znajduje się najbardziej po prawej stronie w drzewie BST – od korzenia poruszamy się ciągle w prawo.

Drzewo BST

- Wstawienie nowego elementu do drzewa BST – znajdujemy mu miejsce tak jak w wyszukiwaniu binarnym (aż dojdziemy do wskaźnika pustego)

```
insert (węzeł *w, x) -> node*
{
        if (w == null) return new node(x);
        if (x < w.info) w.left := insert(w.left, x);
        else if (w.info < x) w.right := insert(w.right, x);
        else w.info := x;
        return w;
}
```

Drzewo BST

- Usunięcie elementu z drzewa BST – znajdujemy miejsce tego elementu tak jak w wyszukiwaniu binarnym (jak dojdziemy do wskaźnika pustego to elementu nie ma w drzewie):
  - gdy element jest w liściu, to odcinam liść;
  - gdy element jest w węźle z jednym następnikiem, to usuwam ze ścieżki ten węzeł;
  - gdy element jest w węźle z dwoma synami, to z prawego poddrzewa usuwamy minimum (albo z lewego poddrzewa usuwamy maksimum) i przenosimy je do tego węzła.

Przeglądanie drzew BST

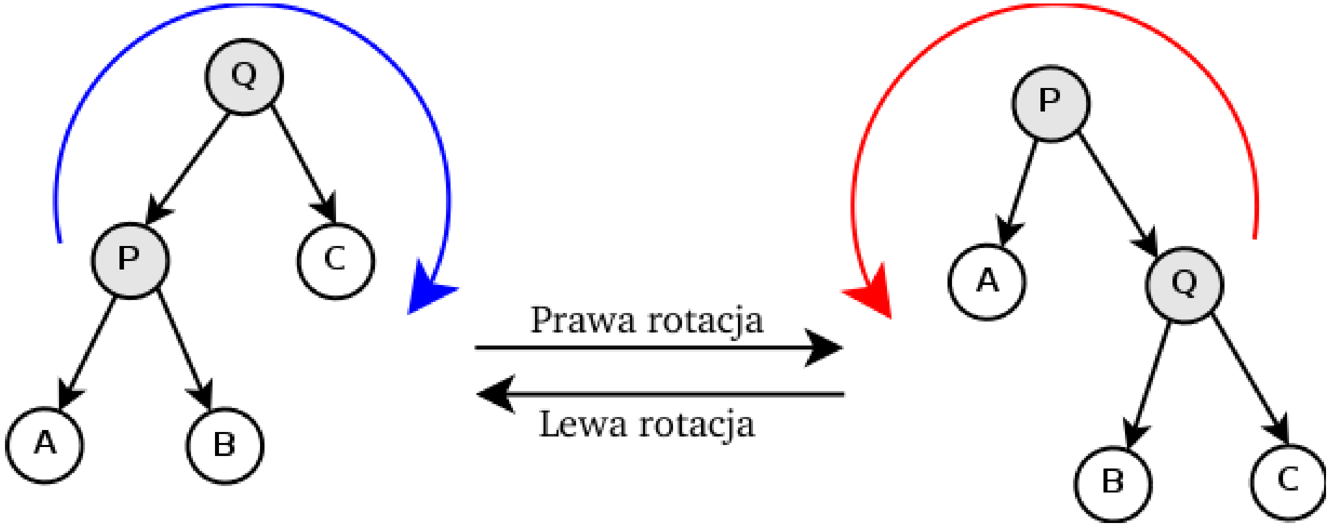
- In-order – najpierw jest przeglądane i przetwarzane lewe poddrzewo w porządku in-order, potem korzeń a na końcu prawe poddrzewo w porządku in-order.
- Przeglądanie drzewa BST w porządku in-order gwarantuje, że węzły tego drzewa zostaną przetworzone w kolejności od najmniejszej do największej wartości.
- Przeglądanie in-order można wykorzystać do sortowania danych.

Przeglądanie drzew BST

- Pre-order – najpierw jest przeglądany i przetwarzany korzeń drzewa, potem lewe poddrzewo w porządku pre-order a na końcu prawe poddrzewo w porządku pre-order.
- Przeglądanie drzewa BST w porządku pre-order gwarantuje, że na początku będą przetworzone węzły z lewej ścieżki tego drzewa w kolejności od korzenia do węzła o najmniejszej wartości.
- Post-order – najpierw jest przeglądane i przetwarzane lewe poddrzewo w porządku post-order, potem prawe poddrzewo w porządku post-order a na końcu korzeń drzewa.
- Przeglądanie drzewa BST w porządku post-order gwarantuje, że na końcu będą przetworzone węzły z prawej ścieżki tego drzewa w kolejności od węzła o największej wartości do korzenia.

Rotacje w drzewie BST

- Rotacja w lewo
- Rotacja w prawo
- W dowolnym wewnętrznym węźle drzewa BST można zrobić rotację (w lewo albo w prawo) i nie zostanie zniszczony porządek symetryczny w tym drzewie.



Drzewa AVL

- Drzewo AVL (nazwa pochodzi od nazwisk rosyjskich matematyków: **A**delsona-**V**elskiego oraz **L**andisa, którzy wymyślili i opublikowali to drzewo w 1962 roku) to drzewo BST, w którym każdy węzeł ma dodatkowy atrybut: balans (2 bity); w każdym węźle drzewa AVL wysokość lewego i prawego poddrzewa każdego węzła różni się co najwyżej o jeden (w prostszej implementacji w węzłach takiego drzewa jest pamiętana wysokość poddrzewa zakorzenionego w danym węźle).
- Wysokość n-elementowego drzewa AVL wynosi O(log n).

Drzewa AVL

- Drzewo AVL jest drzewem zrównoważonym, którego wysokość jest rzędu $\Theta(\log n)$, gdzie n to liczba węzłów w całym drzewie.
- Zrównoważenie drzewa osiąga się, przypisując każdemu węzłowi współczynnik wyważenia, który jest równy różnicy wysokości lewego i prawego poddrzewa. 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.

Drzewa AVL

- Można udowodnić, że n-elementowe drzewo AVL ma wysokość
  $\leq 1.4405 \cdot \log_2(n+1)$, przez co elementarne operacje słownikowe będą wykonywać się w czasie $O(\log n)$.

Drzewa AVL

- Wstawianie do drzewa AVL może odbywać się tak, jak gdyby było ono zwyczajnym drzewem poszukiwań binarnych, następnie zaś są odtwarzane kroki w kierunku korzenia i wykonywane aktualizacje wyważeń węzłów.
- Zmiana wartości współczynnika wyważenia na 0 oznacza, iż wysokość poddrzewa nie została zmieniona. Operacja wstawiania jest wówczas zakończona.
- Zmiana współczynnika wyważenia na 1 lub -1 oznacza, iż poddrzewo, którego korzeniem jest rozważany wierzchołek, zachowało własność drzewa AVL, lecz jego wysokość uległa zwiększeniu. Informacja o tym przekazywana jest do ojca tego węzła.
- Jeśli współczynnik został zmieniony na 2 lub -2, to drzewo straciło własność AVL. Aby ją przywrócić, potrzebna jest rotacja. Maksymalnie będą potrzebne dwie rotacje, po których nie ma potrzeby wykonywania dalszych aktualizacji.

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, które zostaje teraz dzieckiem rodzica wyszukanego elementu. Po jego usunięciu odtwarzana jest ścieżka od rodzica wyszukanego elementu do korzenia, zaś współczynniki wyważenia są aktualizowane.
- 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. W przeciwieństwie do operacji wstawiania wystąpienie rotacji nie musi oznaczać, iż drzewo zostało wyważone. W pesymistycznym przypadku rotacje będą wykonywane aż do korzenia drzewa.

Drzewa AVL

- Wyszukiwanie w drzewie AVL jest wykonywane tak samo jak w niezrównoważonym drzewie BST dzięki serii porównań wartości wyszukiwanej z wierzchołkami, poczynając od korzenia.
- Wynik porównania oraz istnienie poddrzew pozwala stwierdzić, czy element szukany jest wierzchołkiem, znajduje się w lewym bądź prawym poddrzewie, czy też nie należy do drzewa. Zachowanie struktury AVL pozwala jednak na redukcję pesymistycznego czasu wyszukiwania do O(lg n).
- Operacja wyszukiwania nie modyfikuje struktury drzewa, więc nigdy nie pociąga wyważania drzewa (rotacji węzłów).

Drzewo R-B (czerwono-czarne)

- Drzewo czerwono-czarne (ang. red-black tree) to opracowane przez Rudolfa Bayera w 1972 roku drzewo BST, w którym każdy węzeł ma dodatkowy atrybut: kolor (1 bit); wszystkie węzły w drzewie R-B są pokolorowane na czarno albo na czerwono według następujących zasad:
  - każdy węzeł jest czerwony lub czarny;
  - korzeń zawsze jest czarny;
  - każdy liść jest czarny (traktujemy wszystkie wskaźniki puste null jako liście);
  - jeśli węzeł jest czerwony, to jego synowie muszą być czarni ;
  - każda ścieżka z korzenia do liścia zawiera tyle samo czarnych węzłów.

Drzewo R-B (czerwono-czarne)

- Warunki narzucone na drzewa czerwono-czarne gwarantują, że najdłuższa ścieżka od korzenia do liścia będzie co najwyżej dwukrotnie dłuższa, niż najkrótsza – wynika to wprost z ostaniej własności.
- Można udowodnić, że n-elementowe drzewo czerwono-czarne ma wysokość $\leq 2 \cdot \log_2(n+1)$, przez co elementarne operacje słownikowe będą wykonywać się w czasie $O(\log n)$.