Drzewa AVL

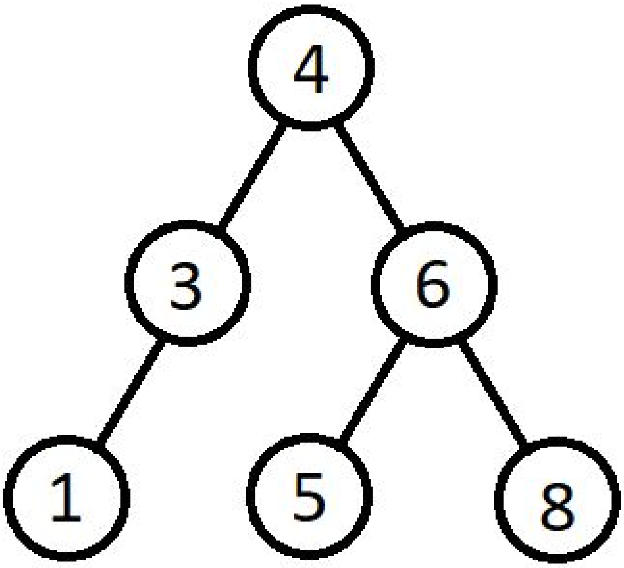
Zuzanna Brzóska

Spis treści

1. Wprowadzenie

1.1. Drzewa binarne oraz BST

Niniejszy referat traktuje o drzewach AVL, które są szczególnym przypadkiem drzew wyszukiwań binarnych, dlatego warto najpierw przypomnieć pojęcie samego drzewa i terminologii z nim związanej.
Drzewo to hierarchiczna, łączona struktura danych, przechowująca dane w wierzchołkach, reprezentująca drzewo matematyczne. Na zbiorze wierzchołków (węzłów) V istnieje pewna relacja S, która gdy zachodzi dla węzłów x i y tak, że jeśli xSy to mówimy, że y jest dzieckiem x.
Rodzicem węzła x nazywamy węzeł y taki, że x jest dzieckiem y.
Węzeł, który nie ma rodzica nazywamy korzeniem (w drzewie jest tylko jeden korzeń), natomiast węzeł bez dzieciek nazywamy liściem.
Ścieżka z węzła x do y to ciąg węzłów, rozpoczynający się od x i kończący na y, taki, że każdy węzeł w ciągu jest w relacji z następnym.
Każda ścieżka, która zaczyna się w korzeniu, a kończy w liściu jest gałęzią.
Wysokość drzewa określamy za pomocą długości jego najdłuższej gałęzi.
Można również wyróżnić pewną część drzewa V, ograniczając się tylko do wybranego węzła wraz z jego dziećmi. Nazywamy tę strukturę poddrzewem V.
Przykładem takiego drzewa jest drzewo genealogiczne.
Drzewo binarne to drzewo, w którym każdy węzeł może mieć co najwyżej jeden prawy i jeden lewy dziecek. Natomiast drzewo wyszukiwań binarnych (BST) to szczególny rodzaj drzewa binarnego, w którego węzłach przechowujemy parami porównywalne elementy, w taki sposób, że dla każdego węzła obiekt przechowywany w nim jest nie mniejszy od obiektów przechowywanych w węzłach lewego poddrzewa i nie większy od obiektów przechowywanych w węzłach prawego poddrzewa tego węzła.
Poprzednikiem węzła x nazywamy węzeł, którego klucz jest bezpośrednio mniejszy od klucza węzła x. Inaczej mówiąc: na drzewie BST nie istnieją węzły, których klucze znajdują się w przedziale pomiędzy kluczem x, a jego poprzednikiem.



Przykładowe drzewo BST

1.2. Problemy związane z najprostszą implementacją drzew binarnych

Drzewa BST w klasycznej, najprostszej postaci mogą się degenerować do list liniowych. Najprostsze algorytmy do wprowadzania/usuwania elementu nie optymalizują kształtu drzewa. Przy n węzłach w najbardziej pesymistycznym przypadku mamy drzewo wysokości n (wtedy drzewo przypomina listę łączoną), a pozytywny przypadek da nam drzewo wysokości $\log_2 n$ (gdy zarówno w prawym i lewym poddrzewie jest mniej więcej tyle samo węzłów). Gdyby na przykład n=1000000 wtedy w pozytywnym przypadku wysokość drzewa będzie rzędu $\log_2(1000000)$ czyli najwyżej 20 - widać zatem, że różnica może być ogromna. Czas operacji wyszukiwania elementów w drzewie binarnego wyszukiwania jest ściśle związany z wysokością drzewa - zależy od niej liniowo. Dlatego wyszukiwanie elementu w drzewie w klasycznej formie może zajmować bardzo dużo czasu. Chcąc rozwiązać ten problem i zoptymalizować czas operacji wykorzystuje się równoważenie drzew binarnych.

1.3. Zrównoważone drzewa binarne

Aby otrzymać pozytywny przypadek czasów operacji drzewo binarnego wyszukiwania powinno mieć taką samą liczbę węzłów w prawym i lewym poddrzewie. Jest tak w przypadku gdy drzewo jest zrównoważone - wtedy różnica wysokości lewego i prawego poddrzewa każdego z węzłów wynosi najwyżej 1. Drzewo nazywamy doskonale zrównoważonym, gdy prawe i lewe poddrzewo różni się najwyżej jednym elementem.
Problem z długim czasem wyszukiwania elementów w drzewie można rozwiązać właśnie przez stosowanie równoważenia drzewa BST, np. algorytmem DSW (jest on jednak czasochłonny - pracuje w czasie liniowo zależnym od ilości węzłów w drzewie, ponieważ najpierw przekształca drzewo w listę, a następnie równoważy się drzewo rotacjami). Aby rozwiązać problem degeneracji drzew do list liniowych stosuje się różne odmiany drzew. Najstarszą z nich jest drzewo AVL.

2. Drzewa AVL

2.1. Drzewa AVL, ich niezmiennik oraz gwarancje czasów operacji

Drzewa AVL (inaczej zwane drzewami dopuszczalnymi) to zrównoważone drzewa poszukiwań, w których zawsze wysokość lewego i prawego poddrzewa każdego węzła różni się najwyżej o 1. Zrównoważenie drzewa osiąga się poprzez przypisanie każdemu węzłowi współczynnika wyważenia (równowagi) bf, który określa różnicę wysokości lewego i prawego poddrzewa - przyjmuje wartości 0 (gdy oba poddrzewa są równej wysokości), 1 (gdy lewe poddrzewo jest wyższe o jeden od prawego) lub -1 (gdy prawe drzewo jest wyższe od lewego o jeden). Są to jedyne dozwolone wartości współczynnika bf w drzewie AVL - to ograniczenie nazywa się własnością drzewa AVL. Aby to osiągnąć po operacjach wstawiania i usuwania elementu równoważy się drzewo poprzez odpowiednie rotacje węzłów drzewa na ścieżce w kierunku korzenia, jeśli trafimy na bf równe 2 lub -2.
Koszt modyfikacji drzewa AVL jest nieco większy niż klasycznego drzewa BST, jednak dzięki własności drzewa AVL mamy zagwarantowany czas wyszukiwania elementu w drzewie o n węzłach rzędu $O(\log_2 n)$ - pesymistyczny czas to $1.44(\log_2 n)$.

2.2. Rotacje w drzewach AVL

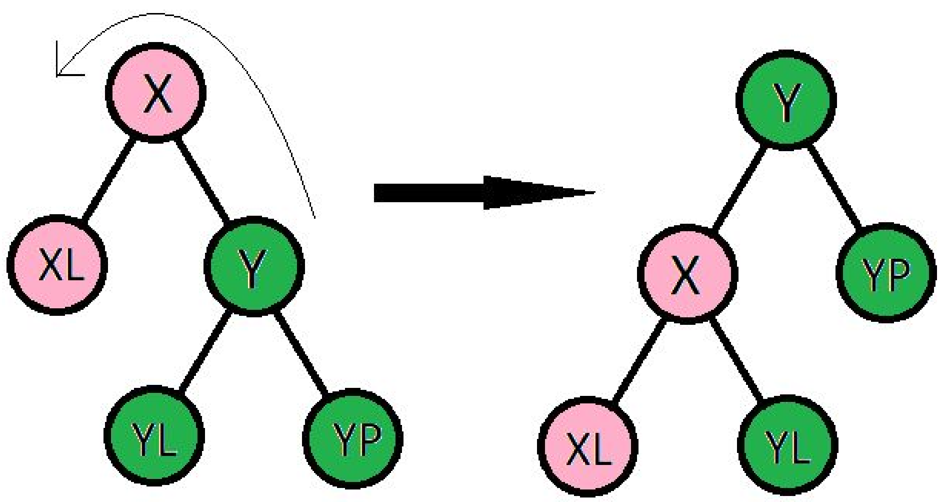
Operacje wstawiania i usuwania węzłów mogą zmieniać wysokość poddrzew, a to może prowadzić do naruszenia własności drzewa AVL (tzn współczynnik bf przyjmuje niedozwoloną wartość 2 lub -2). Równowagę przywracamy wtedy za pomocą rotacji, czyli odpowiedniego przemieszczenia 3 węzłów. Istnieją 4 rodzaje rotacji:

I. rotacja pojedyncza RR
II. rotacja pojedyncza LL
III. rotacja podwójna RL
IV. rotacja podwójna LR

Oznaczenia te określają sposób połączenia węzłów przed wykonaniem rotacji.

I. rotacja RR

W rotacji biorą udział dwa węzły - X oraz Y. Węzeł Y zajmuje miejsce węzła X. Węzeł X staje się lewym dzieckiem Y. natomiast lewy dziecek węzła Y staje się prawym dzieckiem X.



Rotację RR przeprowadzamy gdy węzeł X ma współczynnik bf równy -2 natomiast Y ma współczynnik bf równy 0 lub -1.

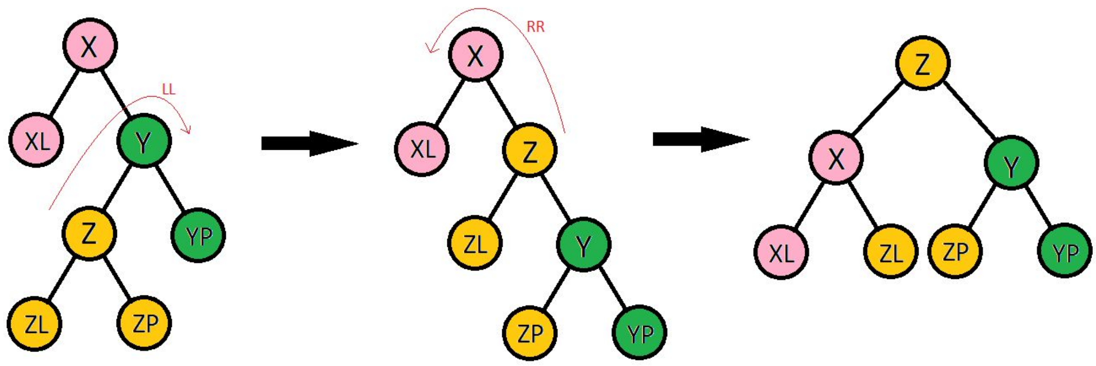
II. rotacja LL

Rotacja LL jest lustrzanym odbiciem rotacji RR. Analogicznie jak wcześniej: mamy węzły X i Y. Węzeł Y zajmuje miejsce X, węzeł X staje się prawym dzieckiem Y, a prawy dziecek Y staje się lewym dzieckiem X.

Rotację LL przeprowadzamy gdy węzeł X ma współczynnik bf równy 2 natomiast Y ma współczynnik bf równy 0 lub 1.

III. rotacja RL

Jest to złożenie rotacji LL i RR. Tym razem mamy trzy węzły: X, Y i Z. Pierwsza rotacja LL przekształca gałąź tak, aby można było przeprowadzić rotację RR, jak opisane wcześniej.



Można jednak przekształcić pozycje węzłów do pozycji końcowej w następujący sposób: węzeł Z zastępuje X. Węzeł X staje się lewym dzieckiem i przejmuje lewego dziecka węzła Z w miejsce swojego prawego. Węzeł Y zajmuje miejsce prawego dziecka Z i przejmuje prawego dziecka Z w miejsce swojego lewego.

Rotację RL przeprowadzamy gdy węzeł X ma współczynnik bf równy -2 natomiast Y ma współczynnik bf równy 1.

IV. rotacja LR

Tak jak wcześniej przy rotacjach pojedynczych, rotacja LR jest lustrzanym odbiciem rotacji RL. Zatem postępowanie jest analogiczne: węzeł Z zastępuje węzeł X, węzeł X staje się lewym dzieckiem Z i przejmuje jego lewego dziecka w miejsce swojego prawego. Węzeł Y staje się prawym dzieckiem Z i przejmuje prawego dziecka Z w miejsce swojego lewego.

Rotację LR przeprowadzamy gdy węzeł X ma współczynnik bf równy 2 natomiast Y ma współczynnik bf równy -1.

2.3. Algorytmy wstawiania i usuwania elementów w drzewie AVL

Operacja wstawiania węzła do drzewa AVL jest dwuetapowa. Najpierw wstawiamy węzeł widentyczny sposób jak w zwyczajnym drzewie BST. Polega to na rekurencyjnym używaniu funkcji wstawiania czyli przechodzeniu przez drzewo w dół i sprawdzaniu czy nasz element jest większy od danego węzła (wtedy kierujemy się na prawo w dół) czy mniejszy (wtedy kierujemy się w dół na lewo). Na koniec wstawiamy element na puste miejsce za ostatnim liściem na jaki trafimy poruszając się tym sposobem. Ta operacja może zmienić własność drzewa AVL dla któregoś z węzłów na tej gałęzi, powyżej miejsca gdzie wstawiliśmy nowy węzeł. Następnie, idziemy od wstawionego węzła w górę do korzenia, aktualizujemy i sprawdzamy współczynniki bf napotkanych węzłów. Jeśli współczynnik jest zamieniony na 0, oznacza to, że wysokość drzewa się nie zmieniła, można wtedy

zakończyć sprawdzanie. Kiedy trafimy na współczynnik bf równy 2 lub -2 musimy zastosować operację rotacji, aby zrównoważyć drzewo.
Usuwanie węzła z drzewa AVL jest bardziej skomplikowane. Trzeba dokładniej rozważyć kilka możliwych przypadków, zarówno co do samego usuwanego węzła jak i do wszystkich węzłów na gałęzi nad nim. Przy usuwaniu tak jak przy wstawianiu jeśli współczynnik przyjmie wartość 2 lub -3 to należy dokonać rotacji, ale inaczej niż poprzednio jest w przypadku zamiany bf na wartość 0 - to oznacza, że poddrzewo uległo skróceniu, a to może wpływać na wyższe partie drzewa, dlatego należy kontynuować sprawdzanie.
Zauważmy, że jeśli węzeł ma oba dziecka, to jego poprzednikiem będzie największy (najbardziej na prawo) węzeł z lewego poddrzewa naszego węzła. Operacja usuwania zawsze zwraca usuwany element, ponieważ jest wykorzystywana rekurencyjnie.
Załóżmy, że chcemy usunąć z drzewa węzeł X.

I. jeśli węzeł X nie posiada potomków:
   Po prostu usuwamy węzeł X.
II. jeśli węzeł X posiada jeden dziecek:
   Zastępujemy węzeł X jego dzieckiem.
III. jeśli węzeł X posiada dwa dziecka:
   Zastępujemy X jego poprzednikiem, a na samym poprzedniku wykonujemy operację usuwania, działamy w sposób rekurencyjny.

Usuwając węzeł w ten sposób własność drzewa AVL mogła zostać naruszona, dlatego teraz należy to sprawdzić na całej gałęzi od usuwanego węzła do korzenia i w razie potrzeby zrównoważyć drzewo rotacjami.

2.4. Alternatywy dla drzew AVL

Inną odmianą drzewa zrównoważonego jest drzewo czerwono-czarne. W drzewie czerwono czarnym nie mamy współczynnik bf, za to mamy do każdego węzła przypisany kolor - czerwony lub czarny. Drzewo czerwono-czarne musi spełniać następujące wymagania:

I. każdy węzeł jest czerwony lub czarny
II. korzeń jest czarny
III. każdy liść jest czarny
IV. jeśli węzeł jest czerwony, to jego dziecka muszą być czarne
V. każda ścieżka z ustalonego węzła do każdego z jego potomków będących liśćmi ma tyle samo czarnych węzłów

Powyższe warunki gwarantują, że drzewo będzie zrównoważone. Drzewo czerwono-czarne często jest porównywane z drzewem AVL, ponieważ pozwala na wykonanie tych samych operacji, przy takiej samej pesymistycznej złożoności czasowej rzędu $\log_2 n$. Drzewo AVL jest prostsze w implementacji i daje bardziej zrównoważoną strukturę, ale jego operacje są bardziej kosztowne (przywrócenie właściwości drzewa czerwono-czarnego wymaga maksymalnie dwóch rotacji, natomiast w drzewie AVL może to wymagać przejścia całej ścieżki od liścia do korzenia).

3. Podsumowanie

Drzewa AVL są jednym z prostszych rozwiązań problemu utrzymania dobrego drzewa wyszukiwań binarnych i zalet wyszukiwania binarnego. Pozwalają utrzymać złożoność obliczeniową klasy $O(\log_2 n)$ przy używaniu drzew BST, która są bardzo wygodną strukturą do poszukiwań danych.
Pewne trudności sprawiło zrozumienie i zaimplementowanie dość skomplikowanej operacji usuwania elementu z drzewa, należało rozważyć dużo różnych możliwości.

Zaletą było, że przy implementacji innych metod można było skorzystać z operacji dla klasycznych drzew BST. Jednak trzeba było co nieco pozmieniać, żeby dodać do węzłów współczynniki bf. Trudnością było wracanie po gałęzi od korzenia, w naszej podstawowej wersji drzewa BST węzły pamiętały tylko swoje dziecka.

4. Źródła

https://pl.wikipedia.org/wiki/Binarne_drzewo_poszukiwa%C5%84
https://pl.wikipedia.org/wiki/Drzewo_AVL
https://pl.wikipedia.org/wiki/Drzewo_czerwono-czarne
https://eduinf.waw.pl/inf/utils/002_roz/mp002.php
https://eduinf.waw.pl/inf/alg/001_search/0119.php
http://www.math.uni.wroc.pl/~jagiella/p2python/