9a. Podstawowe pojęcia teorii drzew

Grzegorz Kosiorowski

Uniwersytet Ekonomiczny w Krakowie

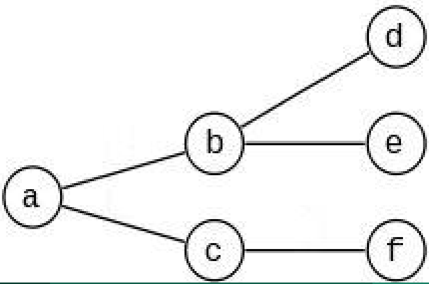
Przypomnienie definicji

Las

Las to graf prosty, acykliczny.

Drzewo

Drzewo to graf prosty, spójny, acykliczny (czyli spójny las). Wierzchołki drzewa nazywamy *węzłami*. Podgraf spójny drzewa nazywamy *poddrzewem*.



Charakteryzacja drzew

Charakteryzacja drzew

Dla grafu $T = (V, E)$ następujące warunki są równoważne:
1. T jest drzewem.
2. T nie zawiera cykli i ma $|V| - 1$ krawędzi.
3. T jest spójny i ma $|V| - 1$ krawędzi (czyli nie da się zmniejszyć liczby krawędzi, by drzewo nadal było spójne).
4. T jest spójny i każda jego krawędź jest mostem.
5. Dowolne dwa wierzchołki grafu są połączone dokładnie jedną drogą prostą.
6. T nie zawiera cykli, lecz dodanie dowolnej nowej krawędzi tworzy dokładnie jeden cykl.

Motywacja

Motywacja badań drzew:

- Struktury danych, czyli przechowywanie informacji umożliwiające szybki dostęp do każdego poziomu struktury.
- Unikanie redundancji.
- Opis zjawisk związanych z hierarchią (starszeństwa, zwierzchnictwa, katalogów, sekwencji czasowej).

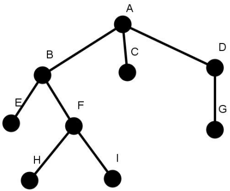
Jako, że najczęściej wierzchołki drzew mają swoje naturalne interpretacje w danym modelu (nazwy, osoby itp.), często przydzielamy im tak zwane „etykiety”, które nieformalnie będziemy rozumieć jako informacje zawarte w tych wierzchołkach.

Drzewo z wyróżnionym korzeniem

Drzewo z wyróżnionym korzeniem

Drzewo z wyróżnionym korzeniem jest to drzewo z wyróżnionym jednym wierzchołkiem, nazywanym *korzeniem*.

Tradycyjnie, drzewa z wyróżnionym korzeniem rysuje się tak, by korzeń był na samej górze. Czasem interpretuje się je jako grafy skierowane ze strzałkami skierowanymi w dół. A jest korzeniem poniższego drzewa.

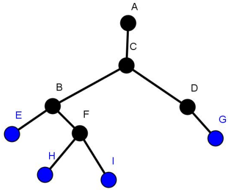


Liście

Liść

Liść drzewa to wierzchołek drzewa stopnia 1, który nie jest wyróżniony jako korzeń.

Niebieskie wierzchołki w tym drzewie to liście. A jest korzeniem, więc choć jest stopnia 1, to nie jest liściem.

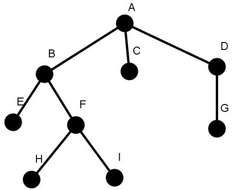


Rodzice i dzieci

Rodzice i dzieci

Jeśli para (v, w) jest krawędzią drzewa z wyróżnionym korzeniem, to v jest *rodzicem* w, a w jest *dzieckiem* v. Ogólniej, w jest *potomkiem* v, jeśli $w \neq v$ i v jest wierzchołkiem jedynej drogi prostej z korzenia do wierzchołka w. v nazywamy wtedy *przodkiem* w.

Wszystkie wierzchołki są potomkami A, ale tylko B, C i D są jego dziećmi. D jest rodzicem G, a B jest przodkiem I.

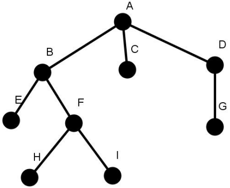


Poziom wierzchołka

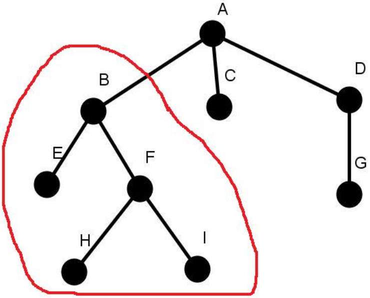
Poziom wierzchołka

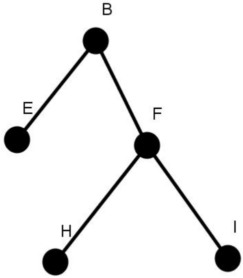
Poziomem wierzchołka v nazywamy długość jedynej drogi prostej od korzenia do v. *Wysokość drzewa* z wyróżnionym korzeniem to największy możliwy poziom wierzchołka tego drzewa.

Poniższe drzewo ma wysokość 3. B, C i D są wierzchołkami poziomu 1, E,F i G - poziomu 2, a H i I - poziomu 3.



Poddrzewo





Drzewo po prawej jest poddrzewem drzewa po lewej o korzeniu B.

Poddrzewo

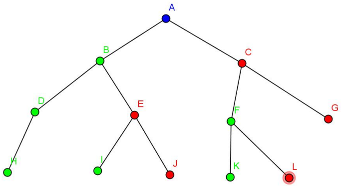
Poddrzewo o korzeniu v jest to drzewo T_v, składające się z korzenia v, wszystkich jego potomków i wszystkich krawędzi (potencjalnie skierowanych) łączących ich.

Przykłady zastosowań drzew z wyróżnionym korzeniem

- Struktura uporządkowania katalogów, podkatalogów i plików
- Opis jakiejkolwiek struktury hierarchicznej (np. władze uczelni)
- Drzewo genealogiczne
- „Drzewkowy" zapis algorytmu postępowania (np. przepis kucharski, poradnik diagnostyczno-medyczny).

Drzewo binarne

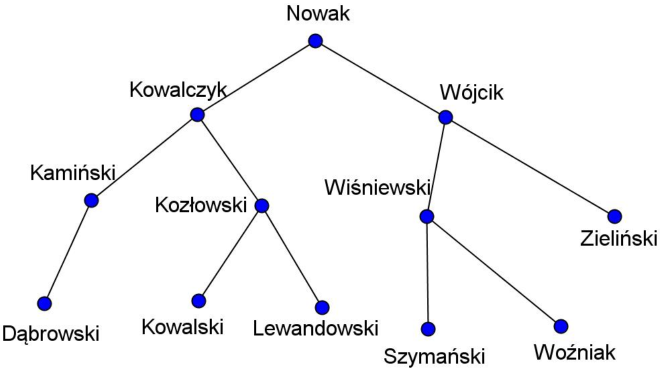
Drzewo binarne jest to drzewo z wyróżnionym korzeniem, w którym każdy węzeł ma **co najwyżej** dwoje dzieci, wśród których wyróżniamy dzieci prawe i lewe.



Zielone wierzchołki to lewe dzieci swoich rodziców, a czerwone - prawe. Warto zauważyć, że wierzchołek D ma tylko dziecko lewe.

Drzewo binarne - przykład

Przykładem drzewa binarnego jest łatwa do przeszukiwania alfabetyczna baza danych (np. pacjentów szpitala), będąca szczególnym przypadkiem tzw. drzewa poszukiwań binarnych.



Zbiór uporządkowany

Poniżej: nie definicja, ale intuicja wystarczającą do naszych zastosowań.

Zbiór uporządkowany

Zbiorem uporządkowanym będziemy nazywać zbiór, w którym każde dwa różne elementy można porównać i powiedzieć, że jeden jest „większy" lub „mniejszy" od drugiego. Sformułowanie „a jest mniejsze od b" oznaczamy zwykle $a \prec b$ lub po prostu $a < b$. Porządek taki powinien być przechodni, czyli zachodzi $(a \prec b \wedge b \prec c) \Rightarrow a \prec c$.

Przykłady zbiorów uporządkowanych: zbiór liczb rzeczywistych ze „standardową" nierównością liczb $<$, zbiór słów z porządkiem alfabetycznym (słowa znajdujące się wcześniej w porządku alfabetycznym są „mniejsze"), zbiór ludzi uporządkowany w kolejności starszeństwa wieku (osoby młodsze są „mniejsze").

Drzewo poszukiwań binarnych

Drzewo poszukiwań binarnych

Drzewo poszukiwań binarnych jest to drzewo binarne składające się z węzłów z uporządkowanymi etykietami (węzłów indeksowanych zbiorem uporządkowanym), w którym lewe poddrzewo każdego węzła zawiera wyłącznie węzły o etykietach nie większych niż etykieta węzła a prawe poddrzewo zawiera wyłącznie węzły o etykietach nie mniejszych niż etykieta węzła.

W ten sposób tworzy się bazy danych, których etykiety są np. liczbami lub słowami (uporządkowanymi alfabetycznie), i które bardzo szybko można przeszukać i znaleźć dany węzeł.

Drzewo poszukiwań binarnych - przeszukiwanie

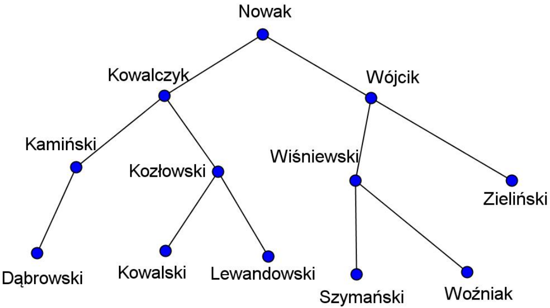
Przeszukiwanie drzewa poszukiwań binarnych jest rekurencyjne:

PRZESZUKIWANIE(v,x)

Dane: Drzewo poszukiwań binarnych z korzeniem v, x - poszukiwana wartość etykiety.
Zmienne: w - węzeł.

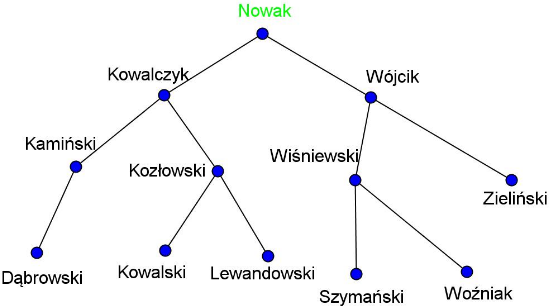
- I. Podstawiamy $w = v$.
- II. Porównujemy etykietę w i x. Jeśli etykiety są równe: STOP.
- III. Jeśli etykieta v jest większa od x, to $w :=$ lewe dziecko v i przechodzimy do kroku V.
- IV. Jeśli etykieta v jest mniejsza od x, to $w :=$ prawe dziecko v i przechodzimy do kroku V.
- V. Wykonujemy PRZESZUKIWANIE(w,x).
- **Rezultat**: w to węzeł o etykiecie x.

Przeszukiwanie - przykład



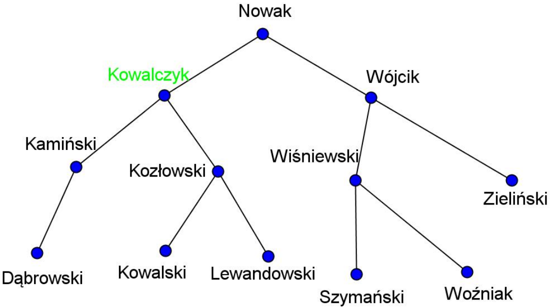
Załóżmy, że za pomocą powyższego algorytmu szukamy w bazie opisanej powyższym drzewem etykiety „Kowalski".
Wykonujemy wtedy algorytm PRZESZUKIWANIE(Nowak,Kowalski).

Przeszukiwanie - przykład



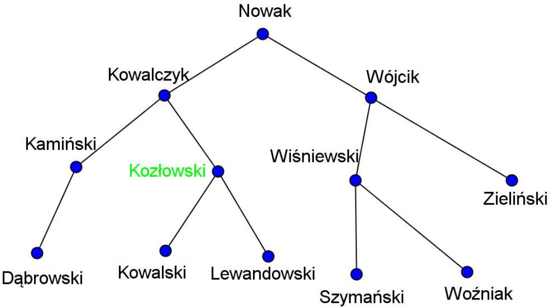
Nowak $\neq$ Kowalski, więc nie kończymy algorytmu. Ponieważ alfabetycznie Kowalski<Nowak, przesuwamy się do lewego dziecka Nowaka, którym jest Kowalczyk i odpalamy algorytm PRZESZUKIWANIE(Kowalczyk,Kowalski).

Przeszukiwanie - przykład



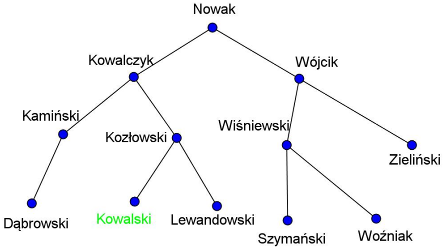
Kowalczyk $\neq$ Kowalski, więc nie kończymy algorytmu. Ponieważ alfabetycznie Kowalczyk<Kowalski, przesuwamy się do prawego dziecka Kowalczyka, którym jest Kozłowski i odpalamy algorytm PRZESZUKIWANIE(Kozłowski,Kowalski).

Przeszukiwanie - przykład



Kozłowski $\neq$ Kowalski, więc nie kończymy algorytmu. Ponieważ alfabetycznie Kowalski<Kozłowski, przesuwamy się do lewego dziecka Kozłowskiego, którym jest Kowalski i odpalamy algorytm PRZESZUKIWANIE(Kowalski,Kowalski).

Przeszukiwanie - przykład



Kowalski = Kowalski, więc algorytm kończy działanie - znalazł szukany obiekt. Warto zauważyć, że stało się to w czwartym kroku i, że każdy wierzchołek tego grafu w 4 krokach można wyszukać tym algorytmem. Gdyby np. sprawdzać wierzchołki po kolei, alfabetycznie, większość wierzchołków znaleźlibyśmy później (do 12 kroków).

Algorytm przeszukiwania - dodatkowe informacje

Jeśli drzewo przeszukiwane jest „dobrze wyważone" (czyli skonstruowane tak, by miało jak najmniejszą wysokość) to czas działania tego algorymu jest $O(\log n)$ (gdzie n jest liczbą wierzchołków).
Algorytmy do samodzielnego stworzenia i przeanalizowania:

- Znajdowanie największego/najmniejszego elementu w drzewie poszukiwań binarnych.
- Znajdowanie poprzednika/następnika danego elementu w drzewie poszukiwań binarnych.
- Wstawianie nowego elementu/usuwanie jednego z elementów, by nie zaburzyć struktury tego drzewa (jak najmniej zmienić wysokość).