Grafy i Zastosowania

3: Drzewa

©Marcin Sydow

Spis zagadnień

- drzewo i las: definicja, charakteryzacje, własności
- kodowanie Prüfera i zliczanie drzew etykietowanych (tw. Cayleya)
- drzewa ukorzenione
- drzewa binarne
- zliczanie drzew binarnych (tw. Catalana)
- zastosowania drzew ukorzenionych w informatyce (BST, kopce, drzewo Huffmana, etc.)

Drzewa

drzewo to graf spójny i bez cykli (acykliczny)

las to graf bez cykli (niekoniecznie spójny)

przykłady

liść to taki wierzchołek drzewa, który ma stopień 1
pozostałe wierzchołki nazywamy **wewnętrznymi**

uwaga: wierzchołki w drzewach często nazywane są też **węzłami** drzewa

Fakt:
Problem izomorfizmu dla drzew nie jest trudny: istnieje szybki (liniowy) algorytm sprawdzający izomorficzność dwóch drzew (Aho et. al. "Algorytmy i Struktury Danych")

Charakteryzacja drzew

Następujące warunki są równoważne :

- graf T jest drzewem o n wierzchołkach
- T ma n-1 krawędzi i jest acykliczny
- T jest spójny i ma n-1 krawędzi
- T jest spójny i każda krawędź jest mostem
- każde dwa wierzchołki T sa połączone dokładnie jedną drogą elementarną
- T jest acykliczny, ale po dodaniu jakiejkolwiek krawędzi powstaje dokładnie jeden nowy cykl

Inne proste własności drzew i lasów *

- każde drzewo posiadające krawędź ma co najmniej 2 liście
- las o dokładnie k składowych ma dokłanie n-k krawędzi.
- w drzewie o conajmniej 3 wierzchołkach, jeśli v jest centralny to $deg(v) \geq 2$
- centrum drzewa to albo 1 wierzchołek albo graf P_2 (tw. Jordana)
- przy ustalonej liczbie wierzchołków n drzewa, P_n ma najmniej liści (2), a $K_{1,n-1}$ (gwiazda) najwięcej (n-1)

Indeks Wienera *

Indeks Wienera $ds(G)$ dla grafu G: suma wszystkich odległości między parami wierzchołków grafu.
$(ds(G) = \sum_{v,w \in V(G)} d(v,w))$

przykład

Fakt:
Przy ustalonej liczbie wierzchołków drzewa, indeks Wienera osiąga:

- wartość minimalną dla grafu będącego gwiazdą: $K_{1,n-1}$ $((n-1)^2)$
- wartość maksymalną dla grafu ścieżkowego: P_n

Ciąg stopni drzewa *

Ciąg liczb dodatnich naturalnych nazwiemy **drzewowym** jeśli pewna jego permutacja stanowi ciąg stopni pewnego drzewa.

przykład

Twierdzenie:
Ciąg $(d_1, \ldots, d_n)$ (dla $n \geq 2$) jest drzewowy $\Leftrightarrow$ $\sum_{i=1}^{n} d_i = 2n - 2$

(uwaga: oczywiście nieco ciekawsza jest implikacja w lewo)

przykład

Drzewa jako podgrafy *

Twierdzenie:
Jeśli T jest drzewem o n wierzchołkach i G jest prostym grafem o minimalnym stopniu $\delta_{min} \geq n - 1$ to wtedy T jest izomorficzne z pewnym podgrafem G.

przykład

Drzewa etykietowane

W tej sekcji przez **drzewa etykietowane** rozumiemy drzewa, których wierzchołki mają etykiety będące kolejnymi dodatnimi liczbami naturalnymi.

przykład

Ile jest różnych drzew etykietowanych o n wierzchołkach?

Zagadnienie to ma ważne zastosowania np. w chemii (gdzie drzewa reprezentują pewne cząsteczki chemiczne, np. alkany C_nH_{2n+2}, oryginalnie zliczane przez Arthura Cayleya w XIX w.)

Kodowanie Prüfera

Kodowanie Prüfera pozwala w jednoznaczny sposób zakodować dowolne drzewo etykietowane o n wierzchołkach za pomocą $(n-2)$-elementowego ciągu liczb z zakresu $[1, n]$.

Algorytm kodowania jest następujący:
Mając dane drzewo etykietowane, niech b_1 będzie najmniejszą liczbą przypisaną liściowi i niech a_1 będzie (jedynym) sąsiadem tego liścia. Usuwamy ten liść z drzewa (wraz z incydentną krawędzią) i zapisujemy a_1 jako pierwszy element ciągu. W powstałym drzewie analogicznie poszukujemy najmniejszej etykiety liścia b_2 i zapisujemy jego sąsiada a_2 jako następny element ciągu a liść usuwamy (wraz z krawędzią incydentną). Postępujęmy tak aż do uzyskania dokładnie $(n-2)$-elementego ciągu: $(a_1, \ldots, a_{n-2})$.
przykład

Dekodowanie Prüfera

Co najważniejsze, każdy $(n-2)$-elementowy ciąg liczb z zakresu $[1, n]$ można "odkodować" używając dekodowania Prüfera i otrzymując n-wierzchołkowe drzewo etykietowane, przy czym dekodowanie jest różnowartościowe

Algorytm dekodowania jest następujący:
Mając dany ciąg $(a_1, \ldots, a_{n-2})$, znajdujemy najmniejszą liczbę $b_1 \in [1, n]$, która nie występuje w ciągu i łączymy krawędzią wierzchołki a_1 i b_1. Usuwamy a_1 z ciągu a b_1 pomijamy w dalszych rozważaniach. Kontynuujemy analogicznie postępowanie dodając kolejne wierzchołki i krawędzie do grafu, aż do wyczerpania elementów ciągu. Na końcu łączymy dwa wierzchołki oznaczone dwoma etykietami nie użytymi dotychczas. W ten sposób otrzymujemy odkodowane drzewo.

przykład

Zliczanie drzew etykietowanych: Twierdzenie Cayleya

Twierdzenie (Cayley, 1989):

Istnieje n^{n-2} różnych n-wierzchołkowych drzew etykietowanych.

dowód: wynika z wzajemnej jednoznaczności kodowania i dedokowania Prüfera, gdyż dokładnie tyle istnieje różnych ciągów o takiej formie

przykład

Wniosek: Graf pełny K_n ma dokładnie n^{n-2} drzew rozpinających

Zliczanie drzew etykietowanych o ustalonym ciągu stopni

Twierdzenie:
Liczba nieizomorficznych n-wierzchołkowych drzew etykietowanych, takich że $deg(v_i) = d_i$ dla $i \in [1, n]$ wynosi:

$$\binom{n-2}{d_1 - 1 \quad \ldots \quad d_n - 1} = \frac{(n-2)!}{(d_1 - 1)! \ldots (d_n - 1)!}$$

(współczynnik wielomianowy)

przykład

Drzewo ukorzenione *

Drzewo ukorzenione to drzewo z pewnym wyróżnionym wierzchołkiem nazywanym **korzeniem** drzewa.

głębokość (albo **poziom**) $depth(v)$ wierzchołka w drzewie ukorzenionym to jego odległość od korzenia.

przykład

Wyodrębnienie korzenia wprowadza w drzewie naturalną hierarchię w zbiorze wszystkich wierzchołków: poziom wierzchołka wyznacza jego miejsce w hierarchii

rysunek drzewa ukorzenionego: korzeń umieszczany jest na górze a kolejne wierzchołki kolejno poziomami poniżej, każdy poziom na tej samej wysokości

przykład

Drzewa ukorzenione, c.d. *

wysokość drzewa ukorzenionego: maksymalna głębokość

przodkiem wierzchołka v nazywamy wierzchołek w leżący na drodze od korzenia do v, v nazywamy wtedy **potomkiem** wierzchołka w (korzeń nie ma przodków a liście nie mają potomków)

przodek w wierzchołka v sąsiadujący z v nazywamy **rodzicem** (ojcem) wierzchołka v, wtedy v nazywamy **dzieckiem** (synem) wierzchołka w

wierzchołek u mający tego samego rodzica co v nazywamy **bliźniakiem** (albo bratem) dla v

liście w drzewie ukorzenionym: wierzchołki bez dzieci

poddrzewo wierzchołka v drzewa ukorzenionego T to podgraf T będący drzewem ukorzenionym, którego korzeniem jest v (jest tyle poddrzew v ile dzieci ma v)

Reprezentacje komputerowe drzewa ukorzenionego *

Najprostsza reprezentacja n-wierzchołkowego, etykietowanego drzewa ukorzenionego w programie komputerowym to tzw. **tablica rodziców**: i-ta pozycja n-elementowej tablicy $n[i]$ zawiera etykietę rodzica wierzchołka o etykiecie i (za wyjątkiem korzenia).

przykład

Stosuje się też stuktury dowiązaniowe o różnych gęstościach powiązań (od pojedynczych, przez podwójne, do wielokrotnych, zależnie od zastosowań)

przykłady

Drzewa $d-$arne (gdzie $d \in N^+$) *

drzewo $d-$arne: drzewo ukorzenione, gdzie każdy wierzchołek ma conajwyżej d dzieci, dla pewnej ustalonej liczby $d \in N^+$

uwaga: za d podstawiamy liczbę, np. "drzewo 3-arne"

zupełne drzewo $d-$arne: drzewo $d-$arne o możliwie dużej liczbie wierzchołków, i wszystkie liście różnią się głębokością conajwyżej o 1

przykłady

drzewo $d-$arne ma co najwyżej d^l wierzchołków na poziomie l.

drzewo $d-$arne o wysokości h zawiera n wierzchołków, gdzie zachodzi: $h+1 \leq \frac{d^{h+1}-1}{d-1}$

(ograniczenie górne to suma ciągu geometrycznego)

Drzewo uporządkowane *

drzewo uporządkowane to drzewo d-arne, gdzie dla każdego wierzchołka wszystkie dzieci mają przypisany pewien porządek liniowy.

przykład

Uwaga: Na **rysunku drzewa uporządkowanego** dzieci rysuje się zgodnie z ich porządkiem od lewej do prawej

porządek standardowy drzewa uporządkowanego to uporządkowanie liniowe wszystkich jego wierzchołków rekurencyjnie leksykograficznie po poziomach a następnie zgodnie z porządkiem dzieci

Uwaga: porządek standardowy odpowiada naturalnemu przeglądaniu rysunku drzewa ukorzenionego od góry do dołu i od lewej do prawej

przykład

Drzewa binarne *

drzewo binarne: 2-arne drzewo uporządkowane, w którym dodatkowo jest określone, które dziecko jest lewe, a które prawe (nie mogą być oba lewe ani oba prawe)

przykład

Obserwacja: pojęcia drzewa, drzewa ukorzenionego, drzewa d-arnego, uporządkowanego, binarnego stanowią hierarchię coraz bardziej specjalizujących się pojęć.

Obserwacja 2: należy odróżniać różne rodzaje izomorfizmów właściwych dla powyższych pojęć (np. dwa drzewa mogą być izomorficzne jako ukorzenione, ale nie jako binarne)

przykłady

Rekurencyjne przeszukiwanie drzew binarnych *

W wielu zastosowaniach ważne jest systematyczne przejrzenie wszystkich wierzchołków (i krawędzi) drzewa binarnego.

Oprócz porządku standardowego (dla dowolnych drzew uporządkowanych) rozróżnia się m.in. pewne 3 specyficzne porządki rekurencyjnego przeglądania drzewa binarnego:

- pre-order (bieżący, lewy, prawy)
- in-order (lewy, bieżący, prawy)
- post-order (lewy, prawy, bieżący)

Każdy z powyższych porządków zaczyna przeszukiwanie od korzenia i rekurencyjnie wywołuje przeszukiwanie w poddrzewach w odpowiedniej kolejności.
(lewy, prawy: oznaczają poddrzewa bieżącego wierzchołka)

przykłady

Geometryczna interpretacja rekurencyjnych przeszukiwań drzew *

Gdy narysujemy drzewo binarne w standardowym porządku, i zakreślimy linię otaczającą drzewo zaczynając ponad korzeniem i obchodząc "naokoło" wszystkie gałęzie przeciwnie do ruchu wskazówek zegara, to mamy następujące interpretacje geometryczne:

- pre-order: wypisz wierzchołek pierwszy raz go napotykając
- post-order: wypisz wierzchołek ostatni raz go napotykając
- in-order: wypisz wierzchołek posiadający lewego syna drugi raz go napotykając, a każdy inny pierwszy raz go napotykając

Zastosowania rekurencyjnych porządków przeszukiwania *

Rekurencyjne podejście do przeszukiwania drzew binarnych upraszcza zapis wielu ważnych algorytmów na tych drzewach, np. obliczanie:

- liczby wierzchołków w drzewie
- wysokości drzewa
- głębokości każdego wierzchołka
- liczby liści, etc.

Zliczanie drzew binarnych *

Twierdzenie:

Liczba b_n różnych drzew binarnych o n wierzchołkach wynosi:

$$b_n = \frac{1}{n+1}\binom{2n}{n}$$

b_n nazywana jest **liczbą Catalana**.

przykład

Zastosowania drzew ukorzenionych *

Drzewa ukorzenione mają liczne zastosowania w:

- informatyce (np. jako struktury danych)
- sztucznej inteligencji (np. drzewa decyzyjne, drzewa gier)
- eksploracji danych (np. kd-drzewa)
- przetwarzaniu języka naturalnego (np. drzewa parsowania)
- organizacji informacji (np. modele hierarchiczne, taksonomie)
- biologii (np. drzewa genealogiczne)
- teorii gier (np. zasada minimaksowa)

Zastosowania drzew ukorzenionych jako struktur danych *

Drzewa ukorzenione mają ważne i liczne zastosowania np. w informatyce jako efektywne implementacje ważnych abstrakcyjnych struktur danych, np:

- drzewo wyszukiwań binarnych (implementacja uporządkowanego słownika) (inne: AVL, B-drzewa, czerwono-czarne, etc.)
- kopiec binarny (implementacja kolejki priorytetowej)
- kopiec dwumianowy (implementacja złączalnej kolejki priorytetowej) (inne: kopiec Fibonacciego, drzewa van Emde Boas, etc.)
- drzewo Huffmana (zwarta reprezentacja kodu prefiksowego)
- drzewo sufiksowe (zwarta reprezentacja przeszukiwanego ciągu symboli)

Przykład: gra NIM

W grze NIM jest 2 graczy, nawijmy ich MIN i MAX.
Zadaniem MIN jest minimalizacja wygranej MAX'a a zadaniem MAX'a jest maksymalizacji swojej wygranej. Przyjmijmy, że wygrana wynosi po prostu 1, gdy wygra MAX a 0, gdy wygra MIN.
Na początku w grze jest n kamyków na jednej kupce. Gracze wykonują ruchy na przemian. Ruch gracza polega na podziale dowolnej leżącej kupki na 2 kupki o *różnych wielkościach*.
Gracz wygrywa, gdy jego rywal nie może wykonać ruchu.

Przykłady rozgrywek

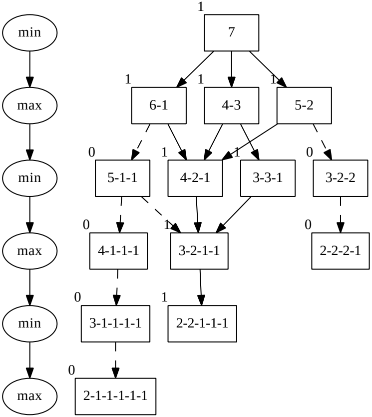
przykład 1: załóżmy, że zaczyna MIN a n=2 (MIN przegrywa)

przykład 2: zaczyna MIN a n=3 (MAX przegrywa)

przykład 3: zaczyna MIN, n=7: czy MIN może wygrać?

Aby odpowiedzieć na to pytanie, można dokonać analizy gry za pomocą **drzewa gry** i tzw. **zasady minimaksowej**.

Drzewo gry NIM dla 7 kamyków



Podsumowanie

Po tym wykładzie wymagana jest przynajmniej:

- znajomość definicji dotyczących różnych rodzajów drzew, lasów i ich elementów oraz własności
- umiejętność wyznaczania kodowania i dekodowania Prüfera
- umiejętność zliczania różnych drzew binarnych, drzew etykietowanych (w tym o ustalonych stopniach wierzchołków)
- znajomość kolejności odwiedzania wierzchołków dla podanego drzewa w omawianych porządkach przeszukiwania (pre/in/post-order)
- umiejętność pisania i analizowania prostych funkcji rekurencyjnych na drzewach binarnych
- znajomość zastosowań drzew w wymienionych dziedzinach

Przykładowe Zadania

Zaproponuj algorytm obliczania *średnicy* danego drzewa nieskierowanego. Przez średnicę rozumiemy maksymalną odległość pomiędzy pewnymi dwoma wierzchołkami tego drzewa

Dziękuję za uwagę