Дерева. Основні означення та властивості. Рекурсія.

Дерева. Основні означення та властивості.

Дерево – в інформатиці та програмуванні одна з найпоширеніших структур даних. Дерева використовують як інструмент обчислень, зручний спосіб збереження даних, їх сортування чи пошуку. Формально дерево визначається як скінченна множина T з однієї або більше вершин (вузлів, nodes), яке задовольняє наступним вимогам:

1. існує один відокремлений вузол — корінь (root) дерева ;
2. інші вузли (за виключенням кореня) розподілені серед $m \geq 0$ непересічних множин T1…Tm і кожна з цих множин в свою чергу є деревом. Дерева T1…Tm мають назву піддерев (subtrees) даного кореня.

В загальному деревом називається ациклічний (тобто без простих циклів) зв'язний граф, а ациклічний незв'язний граф називається лісом. Кістяковим деревом графа називається такий його суграф (підграф, що містить всі вершини графа), що є деревом. Відповідно, кістяковим лісом графа називається пряма сума кістякових дерев усіх компонент зв'язності графа. Кістяковий ліс зв'язного графа складається з єдиного дерева. Кістякові дерева з'явилися в роботах Кірхгофа, який у 1847 р. розробив теорію дерев для визначення сили струму в кожному провіднику та кожному контурі електричної схеми. Пізніше, в 1857 р. Келі розглядав дерева як модель насичених вуглеводнів та розв'язав задачі перерахування дерев. Дерева (головним чином, кореневі орієнтовані з навантаженими вузлами та дугами) широко використовуються в обробці інформації. Досить назвати такі, як пошук та сортування, трансляція, стискання даних та штучний інтелект.

Також можна визначити, що кожна вершина є в свою чергу коренем деякого піддерева. Кількість піддерев вершини має назву ступеня (degree) цієї вершини. Вершина ступеню нуль має назву кінцевої (terminal) або листа (leaf). Некінцева вершина також має назву вершини розгалуження (branch node).

Нехай x — довільна вершина дерева з коренем r. Тоді існує єдиний шлях з r до x. Усі вершини на цьому шляху називаються предками (ancestors) x; якщо деяка вершина y є предком x, то x називається нащадком (descendant) y. Нащадки та предки вершини x, що не збігаються з нею самою, називаються власними нащадками та предками. Кожну вершину x, в свою чергу, можна розглядати як корінь деякого піддерева, елементами якого є вершини-нащадки x.

Якщо вершини x є предком y та не існує вершин поміж ними (тобто x та y з'єднані одним ребром), а також існують предки для x (тобто x не є коренем), то вершина x називається батьком (parent) до y, а y — сином (child) x. Коренева вершина єдина не має батьків.

Вершини, що мають спільного батька, називаються братами (siblings). Вершини, що мають дітей, називаються внутрішніми (internal). Глибиною вершини x називається довжина шляху від кореня до цієї вершини. Максимальна глибина вершин дерева називається висотою.

Якщо існує відносний порядок на піддеревах T1…Tm, то таке дерево називається впорядкованим (ordered tree) або пласким (plane tree). Лісом (forest) - називають множину дерев, які не перетинаються. Найчастіше дерева в інформатиці зображують з коренем, який знаходиться зверху (говорять, що дерево в інформатиці «росте вниз»). Важливим окремим випадком кореневих дерев є бінарні дерева, які широко застосовуються в програмуванні і визначаються як множина вершин, яка має виокремлений корінь та два піддерева (праве та ліве), що не перетинаються, або є пустою множиною вершин (на відміну від звичайного дерева, яке не може бути пустим).

- Кореневий вузол - самий верхній вузол дерева;
- Корінь - одна з вершин, за бажанням спостерігача;
- лист, листовий або термінальний вузол - вузол, який не має дочірніх елементів;
- Внутрішній вузол - будь-який вузол дерева, що має нащадків, і таким чином, який не є листовим вузлом;
- Повний зчеплений ключ - ідентифікатор запису, який утворюється шляхом конкатенації всіх ключів примірників батьківських записів (груп).

Вузли: Вузол є екземпляром одного з двох типів елементів графа, відповідним об'єкту деякої фіксованої природи. Вузол може містити значення, стан або подання окремої інформаційної структури або самого дерева. Кожен вузол дерева має нуль або більше вузлів-нащадків, які розташовуються нижче по дереву (за згодою, дерева 'ростуть' вниз, а не вгору, як це відбувається з справжніми деревами). Вузол, що має нащадка, називається вузлом-батьком щодо свого нащадка (або вузлом-попередником, або старшим). Кожен вузол має не більше одного предка. Висота вузла - це максимальна довжина спадного шляху від цього вузла до самого нижнього вузла (крайовій вузлу), званому листом. Висота кореневого вузла дорівнює висоті всього дерева. Глибина вкладеності вузла дорівнює довжині шляху до кореневого вузла.

Кореневі вузли: Самий верхній вузол дерева називається кореневим вузлом. Мається на увазі відсутність у кореневого вузла предків. Це вузол, на якому починається виконання більшості операцій над деревом (хоча деякі алгоритми починають виконання з «листів» і виконуються, поки не досягнуть кореня). Всі інші вузли можуть бути досягнуті шляхом переходу від кореневого вузла по ребрах (або посиланнях). (Згідно формальному визначенню, кожен подібний шлях повинен бути унікальним). У діаграмах він зазвичай зображується на самій вершині. У деяких деревах, наприклад, купах, кореневий вузол має особливі властивості. Кожен вузол дерева можна розглядати як кореневий вузол піддерева, «зростаючого» з цього вузла.

Піддерева: Піддерево - частина деревоподібної структури даних, яка може бути представлена у вигляді окремого дерева. Будь вузол дерева L разом з усіма його вузлами-нащадками є піддерево дерева T. Для будь-якого вузла піддерева або має бути шлях в кореневий вузол цього піддерева, або сам вузол повинен бути кореневим. Тобто піддерево пов'язано з кореневим вузлом цілим деревом, а відносини піддерева з усіма іншими вузлами визначаються через поняття відповідного піддерева (за аналогією з терміном «відповідне підмножина»).

Упорядкування дерев: Існує два основних типи дерев. У рекурсивному дереві або невпорядкованому дереві має значення лише структура самого дерева без урахування порядку нащадків для кожного вузла. Дерево, в якому заданий порядок (наприклад, кожному ребру, провідному до нащадка, присвоєні різні натуральні числа) називається деревом з іменованими ребрами або впорядкованим деревом зі структурою даних, заданої перед ім'ям і званої структурою даних упорядкованого дерева. Впорядковані дерева є найбільш поширеними серед деревовидних структур. Двійкове дерево пошуку - одне з різновидів упорядкованого дерева.

Подання дерев. Існує безліч різних способів подання дерев. Найбільш загальний спосіб представлення зображує вузли як записи, розташовані в динамічно виділеної пам'яті з дороговказами на своїх нащадків, предків (або і тих і інших), або як елементи масиву, пов'язані між собою відносинами, визначеними їх позиціями в масиві (наприклад, двійкова купа) .

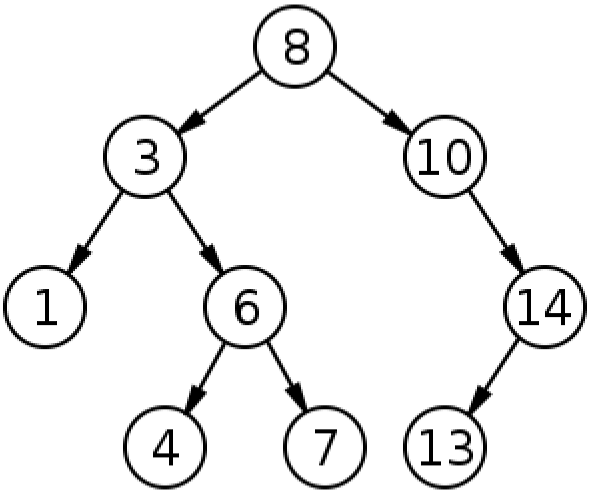


Рис.1 Приклад дерева

Основні операції над деревом:

- перенумерація вершин;
- обхід вершин в різному порядку;
- пошук елемента;
- додавання елемента у визначене місце в дереві;
- видалення елемента;
- видалення цілого фрагмента дерева;
- додавання цілого фрагмента дерева;
- трансформації (повороти) фрагментів дерева;
- знаходження кореня для будь-якої вершини.

Основні означення та властивості:

Теорема. Нехай граф L має n вершин. Тоді такі твердження еквівалентні:

1) граф L – дерево;
2) граф L не містить простих циклів і має (n–1) ребро;
3) граф L зв'язний і має (n–1) ребро;
4) граф L зв'язний, але вилучення довільного ребра робить його незв'язним;
5) довільні дві вершини графу L з'єднані точно одним простим маршрутом;
6) граф L не містить простих циклів, але додавши до нього нове ребро ми отримаємо точно один простий цикл.

<u>Доведення.</u> Вище названу теорему можна довести методом математичної індукції. У разі n = 1 твердження тривіальні. Допустимо, що при n = 1 твердження виконуються. Тоді Доводимо їх для n. Для доведення теореми достатньо доказати наступний ланцюг означень: 1 => 2 => 3 => 4 => 5 => 6 => 1 , так як це означає, що з любого твердження 1 – 6 виводиться будь-яке наступне.

<u>(Відповідно до 1) і 2) означень)</u>. Згідно означенням дерево L не містить простих

циклів . Отже, при вилученні довільного ребра, ми одержимо два графи, у кожен з яких являє собою дерево з меншою, ніж у L, кількістю вершин. Нехай, граф L_1містить n_1вершин, а L_2– n_2: $n_1 + n_2 = n$. За припущенням індукції кількість ребер у кожному з отриманих дерев на 1 менша за кількість вершин, тобто в графі $L_1(n_1 - 1)$ребер, а у графі G2 $(n_2 - 1)$ребер. Отже у графі G було $(n_1 - 1) + (n_2 - 1) + 1 = (n - 1)$ребер.

<u>(Відповідно до 2) і 3) означень).</u> Візьмемо, що граф L незв'язний. Тоді кожна його компонента являє собою зв'язний граф без простих циклів, тобто дерево. Звідси випливає, що кількість вершин у кожній компоненті на одиницю більша від кількості ребер. Отже, загальна кількість вершин графу G більша за кількість ребер принаймні на 2. Але це суперечить тому, що граф має (n–1) ребро.

<u>(Відповідно до 3) і 4) означень).</u> Вилучивши довільне ребро, отримаємо граф з *n* вершинами та *(n–2)* ребрами. Отож, припущення про зв'язність такого графу суперечить теоремі про оцінку (знизу) кількості ребер звичайного графу.

- <u>(Відповідно до 4) і 5) означень).</u> Згідно з тим, що граф L зв'язний, то кожну пару його вершин з'єднано принаймні одним простим маршрутом. Коли якусь пару вершин з'єднано двома простими маршрутами, вони замикаються в простий цикл. Але це суперечить тому, що вилучення довільного ребра робить граф G незв'язним.

<u>(Відповідно до 5) і 6) означень).</u> Припустимо, що граф G містить простий цикл. Тоді довільні дві вершини цього циклу з'єднано принаймні двома простими шляхами, що суперечить твердженню (5). Додавши тепер до графу G ребро е, отримаємо єдиний простий цикл, бо інцидентні ребру е вершини вже з'єднано в графі G точно одним простим шляхом.

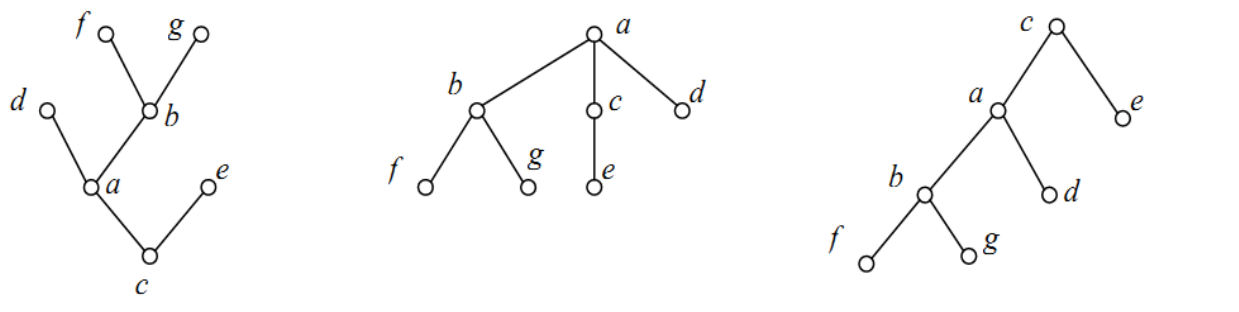
<u>(Відповідно до 6) і 1) означень).</u> Припустимо, що граф G незв'язний. Тоді додавання будь-якого ребра, що з'єднує вершину однієї компоненти з вершиною іншої, не зумовлює утворення простого циклу, що суперечить твердженню (6).

<u>Наслідок з твердження (2).</u> Ліс з *k* дерев, який містить *n* вершин, має *(n–k)* ребер.

У багатьох застосуваннях певну вершину дерева означають як корінь. Тоді можна природно приписати напрямок кожному ребру. Оскільки існує єдиний простий маршрут від кореня до кожної вершини графу, то можна орієнтувати кожне ребро в напрямку від кореня. Отже, дерево разом із виділеним коренем утворює орієнтований граф, який називають кореневим деревом.

Різні способи вибору кореня дають змогу утворити різні кореневі дерева. Наприклад,

- на рис. 2 (а) зображено дерево, а на рис 2 (б,в) – кореневі дерева з коренями відповідно у вершинах а та с.



а б в

Рис. 2

Означення 1.1. Нехай L – кореневе дерево. Якщо *v* – його вершина, відмінна від кореня, то її батьком називають єдину вершину *u* таку, що є орієнтоване ребро *(u, v)*. Якщо *u* – батько, то v – син. По аналогії за генеалогічною термінологією можна визначити інших пращурів і нащадків вершини v. Вершини дерева, які не мають синів, називаються листками. Вершини, які мають синів, називаються внутрішніми. Нехай a – вершина дерева. Тоді піддеревом із коренем a називають підграф, який містить a та всі вершини – нащадки вершини a, а також інцидентні їм ребра.

Означення 1.2. Кореневе дерево називають m-арним, якщо кожна його внутрішня вершина має не більше ніж m синів. Дерево називають повним m-арним, якщо кожна його внутрішня вершина має точно m синів. У разі $m = 2$ дерево називають бінарним.

Означення 1.3. Кореневе дерево, у якому сини кожної внутрішньої вершини впорядковано, називають упорядкованим. Таке дерево зображають так, щоб сини кожної вершини були розміщені зліва направо. Якщо внутрішня вершина впорядкованого бінарного дерева має двох синів, то першого називають лівим, а другого – правим.

Теорема 1.4. Повне m-арне дерево з r внутрішніми вершинами містить $n = mr + 1$ вершин.

Доведення. Кожна вершина, окрім кореня, - син внутрішньої вершини. Оскільки кожна з r внутрішніх вершин має m синів, то всього є, якщо не враховувати корінь, mr вершин, а з урахуванням кореня їх $mr + 1$. ►

Означення 1.5. Рівнем вершини v в кореневому дереві називають довжину простого шляху від кореня до цієї вершини (цей шлях, очевидно, єдиний). Рівень кореня вважають нульовим. Висотою кореневого дерева називають максимальний із рівнів його вершин. Інакше кажучи, висота кореневого дерева – це довжина найдовшого простого шляху від кореня до будь-якої вершини. Повне m-арне дерево, у якого всі листки на одному рівні, називають завершеним.

Кореневе m-арне дерево з висотою h називають збалансованим, якщо всі його листки на рівнях h або $h–1$.

Теорема 1.6. Нехай m-арне дерево має висоту h. Тоді в ньому не більше ніж mh листків.

Доведення. За методом індукції по h. У разі $h=1$ твердження очевидне. Припустимо, що воно справджується для всіх m-арних дерев з висотою $h–1$. Доведемо її для дерева L з висотою h. Тоді його листки – це листки піддерев, отриманих із L вилученням ребер, які з'єднують корінь дерева L з кожною вершиною рівня 1, тобто з кожним сином кореня. Кожне з цих піддерев має не більшу висоту, ніж $h–1$. За індуктивним припущенням всі вони мають не більше ніж m^{h-1}листків. Позаяк таких піддерев не більше ніж m, то загальна кількість листків у дереві L не перевищує mm^{h-1}= m^h. ►

Наслідок. Якщо m-арне дерево з висотою h має t листків, то $[h \geq \log_m t]$. Якщо m-арне дерево повне та збалансоване, то $h = [\log_m t]$. $[x]$– найменше ціле число, яке більше чи дорівнює x.

Доведення. За теоремою 1.6 $t \leq m^h$. Прологарифмуємо цю нерівність за основою m:$\log_m t \leq h$. Оскільки h – ціле, то $h \geq [\log_m t]$. Тепер припустимо, що дерево повне та збалансоване. Вилучимо всі листки на рівні h (разом з інцидентними їм ребрам). Отримаємо завершене m-арне дерево висотою $h–1$. Воно має $m^{(h-1)}$листків. Отже, $m^{(h-1)} < t \leq m^h$. Звідси випливає, що $h - 1 < \log_m t \leq h$, тобто $h = [\log_m t]$. ►

Рекурсія.

Об'єкт називають рекурсивним, якщо він містить сам себе чи його означено за допомогою самого себе. *Рекурсія* – потужний засіб у математичних означеннях.

Означимо повне бінарне дерево рекурсивно (рис.3):

- R (ізольована вершина) – повне бінарне дерево;
- якщо A та B - повні бінарні дерева, то конструкція, зображена на рис. 3. — *повне бінарне дерево.*

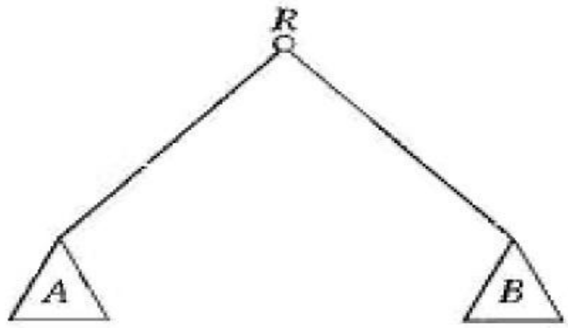


Рис. 3

Рекурсивне означення функції $(n)!$ для невід'ємних цілих чисел має такий вигляд:

- $(0)! = 1;$
- якщо $n > 0$, то $(n)! = n(n-1)!$;

Очевидно, що важливість рекурсії пов’язана з тим, що вона дає змогу означити нескінченну множину об’єктів за допомогою скінченного висловлювання. Так само нескінченні обчислення можна описати за допомогою скінченної рекурсивної програми, навіть якщо вона не містить явних циклів. Проте найдоцільніше використовувати рекурсивні алгоритми тоді, коли розв’язувану задачу, обчислювану функцію чи оброблювані дані задано за допомогою рекурсії.