

ЛЕКЦІЯ 5

НЕЛІНІЙНІ СТРУКТУРИ ДАНИХ

5.1. Дерева

Деревоподібні і взагалі графові структури мають дуже широке застосування. Найчастіше **деревоподібні структури використовуються** у наступних випадках:

- а) при трансляції арифметичних виразів;
- б) при формуванні таблиць символів у трансляторах;
- в) у задачах синтаксичного аналізу;
- г) при трансляції таблиць розв'язків.

При роботі з деревами дуже часто використовуються рекурсивні **алгоритми**, тобто **алгоритми**, які можуть викликати самі себе. При виклику **алгоритму** йому передається в якості параметра вказівник на вершину дерева, яка розглядається як корінь піддерева, що росте із цієї вершини. Якщо вершина термінальна, тобто в неї немає синів, то **алгоритм** просто застосовується до даної вершини. Якщо ж у вершини є сини, то він рекурсивно викликається для кожного із синів. Загальні графові структури застосовують при зображенні розріджених матриць, у машинній графіці, при пошуку інформації і в інших складних задачах. Прикладом графоподібної структури є ієрархічна циклічна структура, що має рівні, аналогічно дереву або орієнтованому графу, але елементи на всякому рівні можуть бути циклічно зв'язані, як, наприклад, у ієрархічних списках. У структурах даних найчастіше використовується табличний або матричний спосіб задання графа.

Кореневим деревом називають орієнтований граф, у якого:

- 1) є одна особлива вершина, в яку не заходить жодне ребро і яку називають коренем дерева;
- 2) у всі інші вершини заходить рівно одне ребро, а виходить скільки завгодно;
- 3) немає циклів.

Існує ще так зване рекурсивне визначення дерева, згідно з яким дерево трактується як скінченна **множина** вершин T , кожна з яких (крім кореня) належить до однієї з підмножин T_1, \dots, T_m ; $m \geq 0$, $T_i \cap T_j = \emptyset$, $i \neq j$. Під**множина** T_i називається піддеревом даної вершини. Число піддерев даної вершини називають**ступеню** цієї вершини. Вершину з нульовою ступенню називають **листом**. Всі інші вершини називаються внутрішніми.

Рівнем або **рангом** вершини по відношенню до дерева називають довжину шляху від кореня до цієї вершини плюс одиниця.

Довжина шляху - це кількість дуг, які треба пройти від кореня для досягнення даної вершини.

Висота дерева дорівнює кількості рівнів у дереві.

Говорять, що кожний корінь є батьком коренів своїх піддерев. Останні є синами свого батька і братами між собою.

Дерево називають n -арним, якщо кожний його вузол має не більше n потомків. На рис.5.1 зображено 3-арне дерево.

Інший приклад деревоподібної структури дають алгебраїчні формули. На рис. 5.2 зображено дерево, що відповідає арифметичному виразу $a - b(c/d + e/f)$. Зв'язок між формулами і деревами дуже важливий для побудови трансляторів арифметичних виразів.

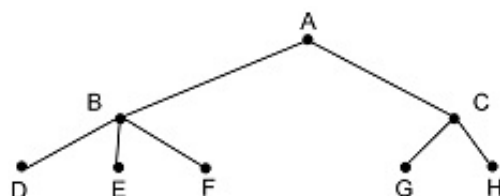


Рис. 5.1. Приклад зображення 3-арного дерева

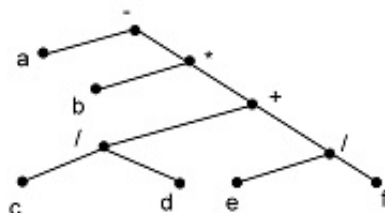


Рис.5.2.Приклад зображення формули у вигляді дерева

5.1.1. Бінарні дерева

Всяке дерево може бути зведено до бінарного. **Бінарним** називають таке 2-арне дерево, в якого один потомок є лівим, а другий - правим. Вершина бінарного дерева може взагалі не мати потомків, або мати тільки ліве, або тільки праве піддерево, або обидва піддерева одночасно.

Бінарне дерево з n вершинами називають **збалансованим**, якщо різниця між рівнями будь-яких двох вершин не більша від одиниці.

Бінарні дерева бувають позиційними і впорядкованими. У **впорядкованих** деревах між множиною вершин і натуральним рядом чисел існує взаємно однозначна відповідність. **Позиційні** дерева відрізняються від впорядкованих тим, що в перших важлива позиція вершини, а в других - тільки номер (рис.5.3).

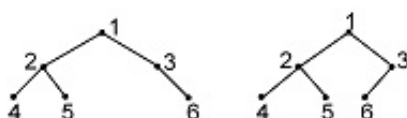


Рис.5.3. Різні позиційні, але однаково впорядковані бінарні дерева

У позиційному бінарному дереві кожна вершина може бути єдиним способом описана за допомогою рядка символів над алфавітом $\{0, 1\}$. Це дає значну зручність для зображення дерев в пам'яті комп'ютера. При цьому корінь дерева характеризується рядком "0". Всякий син вершини Z характеризується рядком, префікс (початкова частина) якого є рядком, що характеризує Z . Рядок, приписаний будь-якій висячій вершині Z , не є префіксом ні для яких рядків, що характеризують інші вершини дерева. Множина рядків, що відповідає висячим вершинам деякого дерева, утворює префіксний код цього дерева (рис.5.4). Довжина рядка, що відповідає вершині, дорівнює рівню цієї вершини.

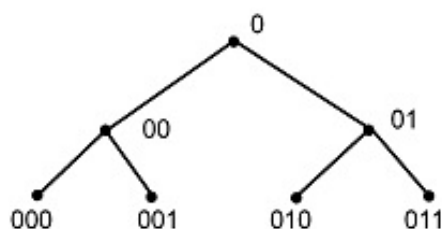


Рис. 5.4. Символьний опис дерева

Багато алгоритмів формальних граматик і пошуку даних мають структуру бінарного дерева. Прикладом є алгоритми обробки префіксних формул у трансляторах арифметичних виразів.

5.1.2. Алгоритм обходу дерева

Алгоритм обходу дерева являє собою спосіб методичного дослідження вершин дерева, при якому кожна вершина проглядається тільки один раз. Повне проходження дерева дає лінійне розміщення вершин, після якого можна говорити про "наступну вершину" як таку, що розміщується або перед даною вершиною, або після неї. Розглянемо три алгоритми обходу бінарного дерева на прикладі дерева, зображеного на рис. 5.5, а.

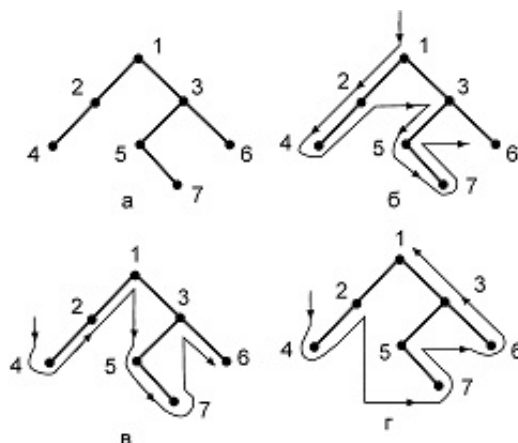


Рис.5.5. Алгоритми обходу бінарного дерева:

а) бінарне дерево; б) низхідний обхід: зверху вниз;

в) змішаний обхід: зліва направо; г) висхідний обхід: знизу – вверх.

1. Послідовність **низхідного обходу**(рис.5.5,б): обробка кореня, низхідний обхід лівого піддерева, низхідний обхід правого піддерева. Вершини дерева проглядаються в послідовності: 1, 2, 4, 3, 5, 7, 6.

2. Послідовність **змішаного обходу** (рис. 5.5,в): змішаний обхід лівого піддерева, обробка кореня, змішаний обхід правого піддерева. Вершини дерева проглядають в послідовності: 4, 2, 1, 5, 7, 3, 6.

3. Послідовність **висхідного обходу**(рис.5.5,г): висхідний обхід лівого піддерева, висхідний обхід правого піддерева, обробка кореня. Вершини дерева проглядаються у послідовності: 4, 2, 7, 5, 6, 3, 1.

5.1.3. Зображення в пам'яті комп'ютера графоподібних структур

Ознайомимось із загальною задачею зображення графа $G=(N,E)$ у вигляді спискової структури, тобто запам'ятовування його на зчепленій пам'яті. Розглянемо орієнтовані графи. Для кожної вершини n існує множина вершин $E(n)$, з якими вона з'єднана ребрами. Тоді один із способів зображення такого графа полягає в тому, щоб кожній вершині співставити одну ділянку списку, в якій одне поле відвести під мітку вершини, якщо така існує, і щонайменше $|E(n)|$ полів - під множину вказівників, тобто ребер. Якщо задано ваги ребер, то потрібно ще додатково $|E(n)|$ полів для зберігання цих ваг. Це один із способів зберігання графів у зчепленій пам'яті.

Але інформацію про ребра можна відділити від інформації про вершини, для чого потрібно ввести два типи ділянок (одну для вершини, другу для ребер) і зв'язати ці ділянки в один

ланцюг. Крім того, всі ребра можна зберігати окремо, по одній ділянці на кожне ребро, і зв'язувати ці ділянки в один ланцюг.

Таким способом зображення графів можна користуватися також для графів з двосторонньою орієнтацією, тоді кожний напрямок треба зображувати незалежно, що створює дублювання інформації, і при зміні графа під час роботи потрібно було б змінювати його в двох місцях. Цю незручність можна ліквідувати за допомогою петель. В цьому випадку для кожного ребра (a, b) потрібна буде зв'язуюча ділянка, в якій зберігався б вказівник, що веде через інші вершини у вершину a , і вказівник, що веде через інші вершини у вершину b , а також додаткові поля для зберігання ваг і міток.

Отже, для зображення в пам'яті графів існує **декілька способів**:

- 1) використовується матриця (або матриця суміжності, або матриця інцидентності), яка зберігається стандартним способом у векторній пам'яті;
- 2) використовується спискова структура у вигляді черги, в якій вказівники відповідають ребрам графа;
- 3) використовується спискова динамічна структура, де для кожної нової підмножини пам'ять виділяється в процесі побудови фрагмента графа.

Кожний з цих способів має свої позитивні і негативні сторони. Який із них більше підходить для зображення графа, залежить від кількості ребер.

Для зображення дерев, як різновиду графів, можна застосовувати ті ж способи, що і для зображення графів взагалі. Однак матриці суміжності, як правило, не використовуються, оскільки вони дуже розріджені. Майже завжди **для зображення дерев застосовують списки**, в яких вказівники зображують ребра.

Дерева можна зберігати в пам'яті також у послідовному вигляді, але при цьому вибирається певний напрямок обходу дерева. Тоді зберігаються тільки вершини дерева, а зв'язки (ребра) опускаються, оскільки порядок розташування вузлів вказує на зв'язок між ними.