Бінарне дерево та його застосування

Граф G називається *деревом*, якщо він є зв'язним і не має циклів (отже, він в тому числі не має петель і кратних ребер). Граф G називається *лісом*, якщо всі його компоненти зв'язності є деревами.

Якщо граф G є деревом, для будь-яких двох різних вершин графа G існує єдиний (і притому простий) ланцюг.

У дереві G число вершин на одиницю більше, ніж число ребер: дерево з n вершинами ($n>1$) завжди містить $m = n - 1$ ребер.

Звернемо увагу ще на дві властивості графа G, який є деревом:

1) граф G не містить циклів, але додавання ребра між двома будь-якими несуміжними вершинами, призводить до появи рівно одного й притому простого циклу;

2) граф G є зв'язним, але він втрачає цю властивість при вилученні будь-якого ребра.

Вершина v дерева G називається *кінцевою*, або *висячою*, або *листом*, якщо її степінь $\rho(v)$ дорівнює одиниці. Інцидентне кінцевій вершині ребро також називається кінцевим.

Якщо скінченне дерево складається більш ніж з однієї вершини, воно має хоча б дві кінцеві вершини і хоча б одне кінцеве ребро.

Нехай у дереві G відзначена вершина v_0. Цю вершину називають *коренем* дерева G, а саме дерево – *деревом з коренем*. У такому дереві можна природним образом орієнтувати ребра (в напрямку від кореня). Орієнтоване в такий спосіб дерево з коренем називається орієнтованим деревом. Якщо змінити вказаний напрямок всіх ребер дерева на протилежний, дістанемо також орієнтоване дерево, в якому всі ребра матимуть напрямок до кореня.

Для порівняння дерев їх приводять до центрально-кореневої форми. Для цього необхідно спочатку знайти центри дерева. Одним із способів зробити це є відсічення висячих вершин: послідовно відсікають вершини 1-го, 2-го, … порядків. Відсічення виконується до тих пір, доки не залишиться або ребро з інцидентними йому двома вершинами максимального порядку, або одна вершина (рис. 4.5). В першому випадку (***а***) дерево має так званий біцентр, і з двох вершин біцентра кореневою вершиною найчастіше вважається та, з якої виходить менше листів; в другому (***б***) залишається єдиний центр, який і вважають коренем дерева.

Відповідну центрально-кореневу форму дерев, поданих на рис. 4.5, бачимо на рис. 4.6.

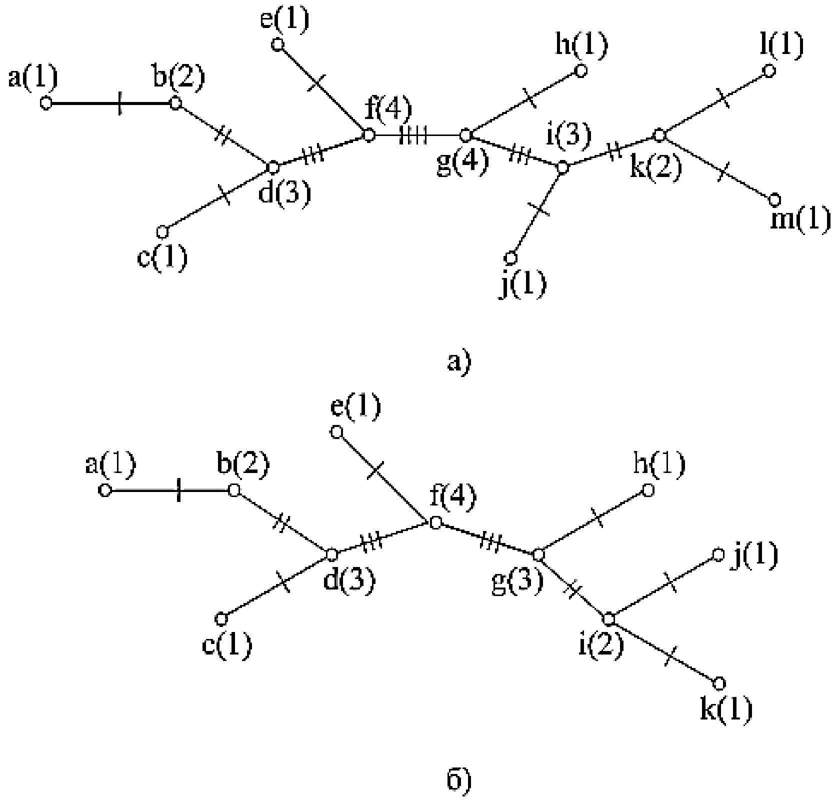

Рис. 4.5

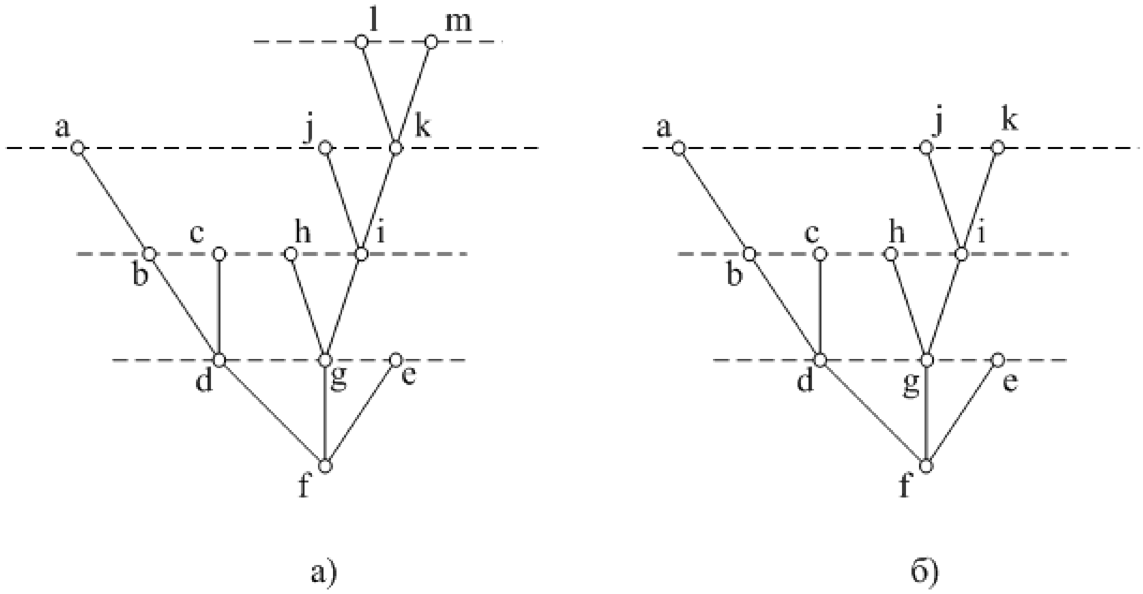

Рис. 4.6

Відносно елементів орієнтованих дерев встановилася відповідна додаткова термінологія. *Нащадок* v вершини u – це вершина v, в яку веде шлях з вершини u; при цьому u називається *предком* для v. Якщо довжина цього шляху дорівнює 1, тобто з u до v веде дуга, то u – *батько* для v, а v – *син* для u. Зрозуміло, що кінцева вершина (лист) не має нащадків. *Висотою* дерева називають довжину найдовшого шляху від кореня до будь-якого листа, *висотою* (або *рівнем*) вершини v – довжину шляху від кореня до цієї вершини. Наприклад, на рис. 4.6,***а*** висота дерева дорівнює 4, вершина k має рівень 3, лист h – рівень 2, корінь f – рівень 0 тощо. Центрально-кореневу форму дерева найзручніше зображувати з урахуванням рівня відповідних вершин, починаючи з кореня, який вважається розташованим на нульовому рівні.

Бінарним (двійковим) деревом T називається орієнтоване дерево, з кожної вершини якого може виходити не більше двох дуг.

Кожна вершина бінарного дерева може мати або двох синів – *лівого* і *правого* (так вершина b на рис. 4.7 має лівого сина a і правого сина c, корінь g має лівого сина d і правого сина j), або мати тільки лівого сина (лівий син h вершини j), або тільки правого сина (вершина i – єдиний правий син вершини h), або не мати жодного сина (листя a, c, e, i). В той же час назвемо вершини a, b, c на рис. 4.7 нащадками вершини d, а вершину d – предком для вершин a, b, c.

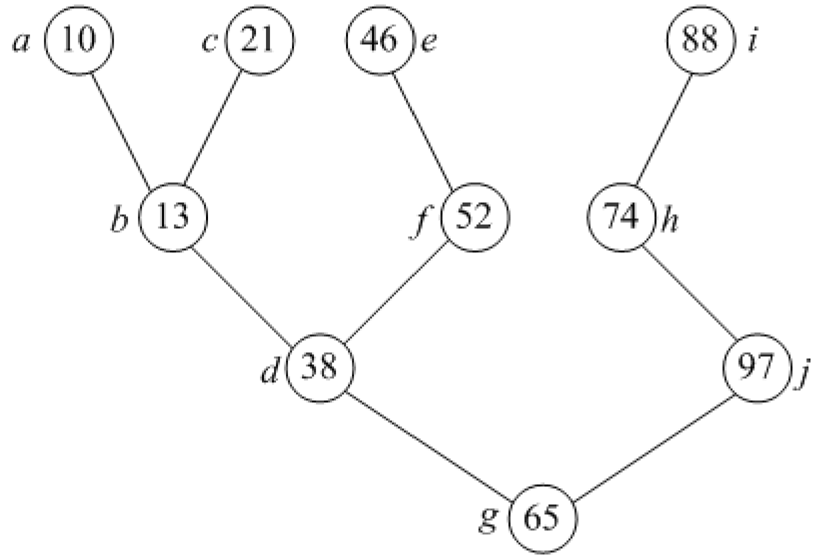

Рис. 4.7

Лівим піддеревом (*лівою гілкою*) вершини v називається максимальне піддерево $T_l(v_l) \subset T$, коренем якого є лівий син v_l вершини v. Відповідно *праве піддерево* $T_r(v_r) \subset T$ (*права гілка*) вершини v визначається як максимальне піддерево з коренем v_r – правим сином вершини v.

Для вершини v двійкового дерева на будь-якому її рівні встановлюється відношення строгого порядку: якщо v_l – лівий син вершини v і v_r – її правий син, то $v_l < v < v_r$. Це відношення дає можливість використовувати двійкове дерево для зберігання довільної інформації в упорядковому виді.

Приклад. До користувача надійшла послідовність десяти чисел: 65, 97, 38, 52, 74, 13, 88, 21, 46, 10 з метою її збереження. Розмістити вказані числа у вершинах двійкового дерева, водночас забезпечивши впорядкування даної послідовності. Після завершення будови дерева скласти алгоритм відновлення з нього послідовності чисел в її упорядкованому вигляді.

Розв'язання. Утворення потрібного дерева починається з розміщення першого числа послідовності (65) у кореневій вершині (рис. 4.7). Після цього у кореня з'являються його лівий та правий сини, яким привласнюються відповідно числа 97 (правому сину, оскільки 97 > 65) і 38 (лівому сину, оскільки 38 < 65). Тим самим у дереві, що створюється, починається формування лівого й правого піддерев. Подальше розміщення чергового числа (a) здійснюється відповідно до такого рекурсивного алгоритму: розміщуване число послідовно порівнюється з кореневими значеннями u_i, які воно проходить, і в залежності від результата цього порівняння воно рухається в напряму лівого піддерева, якщо $a < u_i$, або в напряму правого піддерева, якщо $a > u_i$; цей рух продовжується до тих пір, поки не буде досягнуто лист; тоді для цього листа додається відповідний (лівий або правий) син, в якому й розміщується число a. У табл. 4.7 проілюстровано рух чисел 88 і 21, який вони здійснюють для визначення їхнього місця у дереві.

Таблиця 4.7

a = 88			*a* = 21		
Крок	Визначальне порівняння	Напрям руху	Крок	Визначальне порівняння	Напрям руху
1	88 > 65	Праве піддерево	1	21 < 65	Ліве піддерево
2	88 < 97	Ліве піддерево	2	21 < 38	Ліве піддерево
3	88 > 74	Праве піддерево	3	21 > 10	Праве піддерево

У другій частині завдання потрібно встановити такий порядок обходу вершин побудованого дерева, який встановлював би упорядкований (за зростанням) вивід збережених числових значень. Найкращим для цього є так званий інфіксний (або зворотний) порядок обходу вершин, при якому послідовно здійснюються а) рекурсивний огляд кожного лівого піддерева (починаючи з крайнього лівого листа), б) огляд відповідної кореневої вершини, в) огляд відповідного правого піддерева. Як легко бачити, результатом зазначеного обходу в нашому прикладі є наступна упорядкована числова послідовність: 10, 13, 21, 38, 46, 52, 65, 74, 88, 97. □

Зауваження.

1. Граф G називається *позначеним*, якщо його вершинам привласнені фіксовані позначки. В розглянутому прикладі такими позначками можна вважати сімвольні назви вершин. Таким чином, побудоване дерево є позначеним. Саме завдяки вказаним позначкам забезпечується певний (інфіксний) порядок обходу вершин цього дерева.

2. Крім застосованого у прикладі **інфіксного** порядку обходу вершин дерева існують ще два інші.

Це **префіксний** (або прямий) обхід, при якому здійснюються:

а) огляд кореневої вершини,

б) рекурсивний огляд обох відповідних піддерев,

і **постфіксний** (або кінцевий) обхід, який передбачає:

а) рекурсивний огляд обох піддерев,

б) огляд відповідної кореневої вершини.