Лр№6. 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.

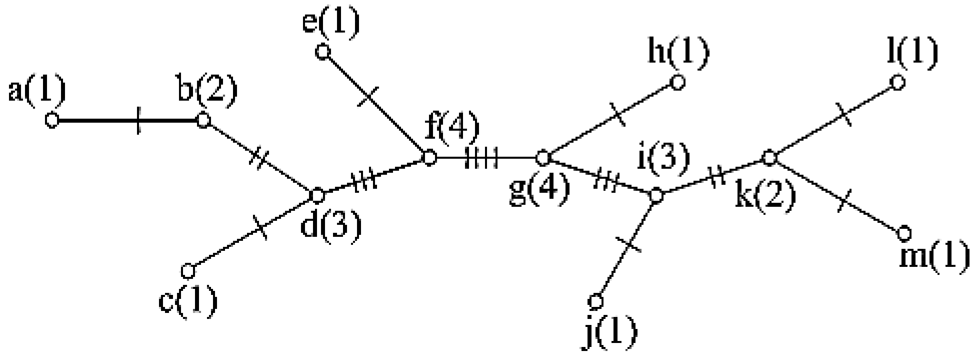
Вершина v дерева G називається *кінцевою*, або *висячою*, або *листом*, якщо її степінь $\rho(v)$ дорівнює одиниці. Інцидентне кінцевій вершині ребро також називається кінцевим.

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

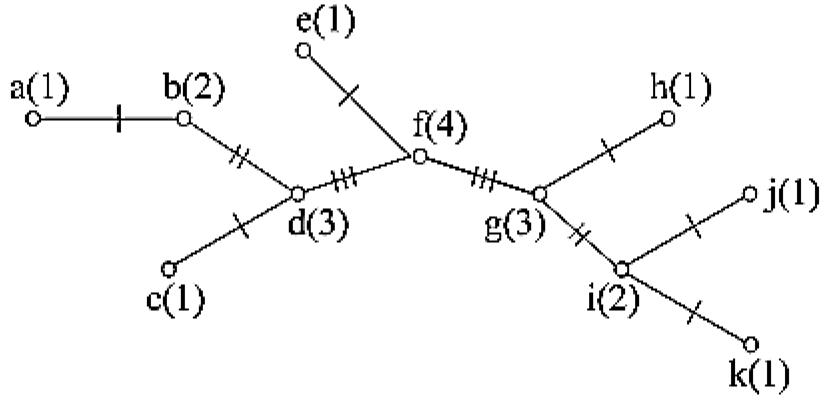
Нехай у дереві G відзначена вершина v_0. Цю вершину називають *коренем* дерева G, а саме дерево – *деревом з коренем*. У такому дереві можна природним образом орієнтувати ребра (в напрямку від кореня). Орієнтоване в такий спосіб дерево з коренем називається орієнтованим деревом. Якщо змінити вказаний 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.



а)



б)

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*.

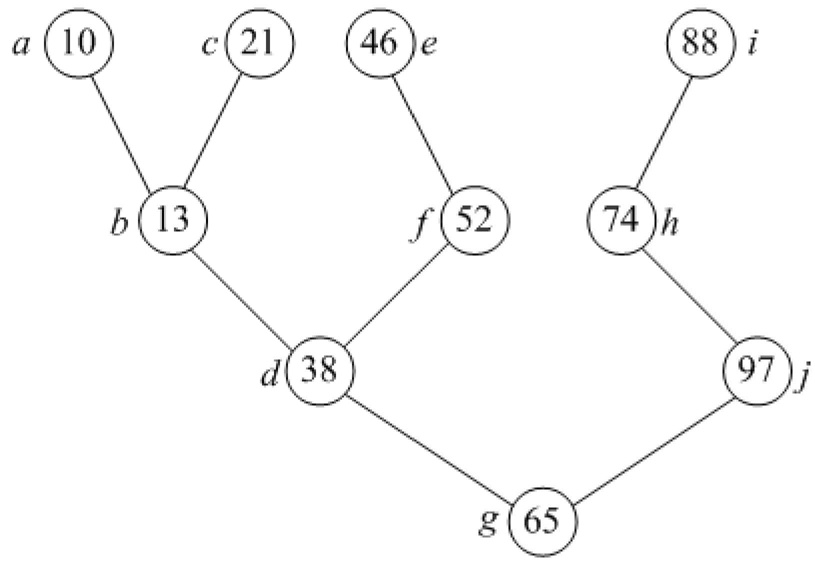


Рис. 4.7

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.

4.2.01. 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.

<u>Розв'язання.</u> 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.

Завдання 1. В задачах 4.2.01 – 4.2.25 упорядкувати за допомогою бінарного дерева задану послідовність чисел.

Зобразити дерево та записати послідовності вершин, які утворюються під час обходу цього дерева: а) в префіксному порядку; б) в інфіксному порядку; в) в постфіксному порядку.

4.2.01. {40, 3, 18, 44, 77, 61, 58, 50, 40, 80, 60, 63, 6, 66, 32}.

4.2.02. {24, 50, 92, 56, 3, 52, 28, 83, 16, 46, 69, 81, 73, 80, 11}.

4.2.03. {41, 13, 32, 85, 91, 3, 70, 96, 44, 80, 13, 60, 65, 92, 8}.

4.2.04. {83, 10, 46, 41, 61, 68, 67, 39, 16, 41, 11, 76, 56, 60, 12}.

4.2.05. {52, 47, 38, 59, 46, 67, 50, 32, 12, 63, 24, 28, 11, 71, 21}.

4.2.06. {75, 14, 81, 20, 21, 96, 85, 64, 66, 15, 46, 34, 7, 96, 91}.

4.2.07. {92, 82, 81, 25, 18, 51, 42, 99, 50, 71, 36, 23, 48, 35, 96}.

4.2.08. {4, 2, 52, 49, 95, 88, 0, 3, 23, 99, 40, 86, 1, 40, 53}.

4.2.09. {53, 4, 81, 86, 15, 5, 11, 55, 77, 51, 56, 31, 84, 8, 86}.

4.2.10. {52, 30, 80, 24, 85, 23, 44, 6, 21, 66, 14, 89, 44, 56, 47}.

4.2.11. {83, 19, 82, 6, 36, 84, 84, 60, 34, 43, 44, 71, 37, 59, 85}.

4.2.12. {33, 58, 77, 7, 65, 95, 89, 22, 75, 56, 99, 73, 28, 30, 57}.

4.2.13. {80, 77, 44, 99, 66, 61, 3, 17, 43, 51, 44, 39, 64, 16, 77}.

4.2.14. {54, 14, 83, 84, 55, 40, 21, 94, 79, 77, 92, 97, 98, 86, 68}.

4.2.15. {70, 59, 3, 70, 95, 23, 16, 20, 95, 40, 90, 28, 56, 97, 59}.

4.2.16. {70, 45, 51, 64, 41, 61, 99, 67, 25, 70, 0, 82, 78, 90, 0}.

4.2.17. {93, 12, 95, 87, 44, 26, 14, 54, 22, 44, 25, 4, 39, 26, 95}.

4.2.18. {76, 88, 25, 27, 32, 78, 33, 30, 82, 90, 49, 80, 40, 33, 64}.

4.2.19. {95, 33, 97, 75, 36, 8, 58, 82, 74, 57, 4, 87, 59, 31, 4}.

4.2.20. {89, 79, 46, 81, 48, 51, 60, 43, 33, 95, 39, 54, 74, 89, 43}.

4.2.21. {39, 58, 92, 35, 14, 11, 50, 11, 57, 0, 52, 23, 8, 24, 91}.

4.2.22. {14, 19, 80, 98, 60, 21, 97, 67, 48, 30, 59, 47, 89, 33, 80}.

4.2.23. {55, 70, 44, 34, 86, 18, 56, 17, 17, 76, 50, 79, 77, 58, 80}.

4.2.24. {47, 3, 46, 22, 43, 43, 88, 76, 37, 42, 29, 0, 94, 17, 61}.

4.2.25. {70, 62, 36, 63, 72, 24, 24, 4, 4, 60, 55, 43, 4, 14, 92}.

В задачах 4.2.01 – 4.2.25 позначене бінарне дерево задано трійками (корінь, ліве піддерево, праве піддерево; головний корінь зверху; знак ^ означає відсутність відповідного піддерева). Зобразити дерево та записати послідовності вершин, які утворюються під час обходу цього дерева: а) в префіксному порядку; б) в інфіксному порядку; в) в постфіксному порядку.

Завдання 2.

4.2.01. (A, B, C), (B, D, E), (C, F, G), (E, H, ^), (F, I, J), (G, K, L), (H, ^, M), (I, N, ^), (K, P, ^).

4.2.02. (A, B, C), (B, ^, D), (C, E, F), (D, G, H), (E, I, J), (F, K, L), (H, ^, M), (I, N, ^), (J, P, ^).

4.2.03. (P, M, N), (M, I, J), (N, K, L), (I, ^, D), (J, E, F), (L, G, H), (E, A, B), (G, C, ^).

4.2.04. (P, M, N), (M, J, K), (N, ^, L), (J, D, E), (K, F, G), (L, H, I), (E, ^, A), (H, B, C).

4.2.05. (A, B, C), (B, D, E), (C, F, G), (D, H, I), (E, J, ^), (F, K, L), (G, M, ^), (H, ^, N), (M, P, ^).

4.2.06. (A, B, C), (B, D, ^), (C, E, F), (D, G, H), (E, ^, I), (F, J, K), (H, L, ^), (I, M, ^), (K, N, P).

4.2.07. (P, M, N), (M, I, J), (N, K, L), (I, C, D), (J, ^, E), (K, F, ^), (L, G, H), (E, A, ^), (F, ^, B).

4.2.08. (P, M, N), (M, J, K), (N, L, ^), (J, ^, E), (K, F, G), (L, H, I), (E, A, B), (G, ^, C), (I, D, ^).

4.2.09. (A, B, C), (B, D, E), (C, F, G), (D, H, ^), (E, ^, I), (F, ^, J), (G, K, ^), (I, L, ^), (J, M, N), (K, ^, P).

4.2.10. (A, B, C), (B, ^, D), (C, E, F), (D, G, H), (E, ^, I), (F, J, ^), (G, ^, K), (H, L, ^), (I, ^, M), (J, N, P).

4.2.11. (P, M, N), (M, I, J), (N, K, L), (I, ^, E), (J, F, ^), (L, G, H), (E, A, B), (F, ^, C), (H, D, ^).

4.2.12. (P, M, N), (M, J, K), (N, L, ^), (J, F, ^), (K, G, H), (L, ^, I), (F, ^, A), (G, ^, B), (H, C, D), (I, ^, E).

4.2.13. (A, B, C), (B, D, E), (C, F, G), (E, H, ^), (F, ^, I), (G, J, ^), (H, K, L), (I, M, N), (J, ^, P).

4.2.14. (A, B, C), (C, D, E), (D, F, G), (E, H, I), (F, ^, J), (G, K, L), (H, M, N), (I, P, ^).

4.2.15. (P, M, N), (M, I, J), (N, K, L), (I, ^, F), (J, ^, G), (L, H, ^), (F, ^, A), (G, B, C), (H, D, E).

4.2.16. (P, M, N), (M, ^, J), (N, K, L), (J, ^, E), (K, F, G), (L, H, I), (E, ^, A), (G, B, ^), (H, ^, C), (I, ^, D).

4.2.17. (A, B, C), (B, D, E), (C, F, ^), (D, ^, G), (E, H, I), (F, ^, J), (G, ^, K), (H, L, M), (I, N, P).

4.2.18. (A, B, C), (B, D, E), (C, F, G), (D, ^, H), (E, I, J), (F, ^, K), (I, L, M), (J, N, P).

4.2.19. (P, M, N), (M, ^, J), (N, K, L), (J, ^, G), (K, ^, H), (L, ^, I), (G, A, B), (H, C, D), (I, E, F).

4.2.20. (P, M, N), (M, I, J), (N, K, L), (I, ^, F), (J, ^, G), (K, ^, H), (F, ^, A), (G, B, C), (H, D, E).

4.2.21. (A, B, C), (B, ^, D), (C, E, F), (D, ^, G), (E, H, I), (F, J, ^), (G, ^, K), (H, ^, L), (I, ^, M), (J, N, P).

4.2.22. (A, B, C), (B, D, E), (C, F, G), (D, ^, H), (E, ^, I), (F, ^, J), (G, ^, K), (H, L, ^), (I, M, N), (K, P, ^).

4.2.23. (P, M, N), (M, ^, J), (N, K, L), (J, E, F), (K, G, H), (L, I, ^), (F, A, B), (I, C, D).

4.2.24. (P, M, N), (M, I, J), (N, K, L), (I, ^, D), (J, ^, E), (K, F, ^), (L, G, H), (D, ^, A), (F, ^, B), (G, ^, C).

4.2.25. (A, B, C), (B, ^, D), (C, E, F), (D, G, H), (E, I, J), (F, K, ^), (G, ^, L), (I, M, ^), (K, N, P).