2.6. ДЕРЕВА: ВИЗНАЧЕННЯ, ТИПИ, АЛГОРИТМИ ОБРОБКИ

Визначення дерева. Корінь та листя дерева. Різні форми представлення дерев. Бінарні дерева. Алгоритми перетворення m-арних дерев у бінарні. Різні алгоритми обходу бінарних дерев. Збалансовані бінарні дерева. Прошиті бінарні дерева. B-дерева

Дерева являють собою окремий клас орієнтованих графів. Ці види орграфів настільки популярні для моделювання різноманітних структур і реалізації алгоритмів, що їх можна розглядати як самостійний вид структур даних, але при цьому будемо спиратися тут на основні положення теорії графів.

Варіант № 1 визначення. Деревоподібна структура з базовим типом *T* визначається як:

– порожня структура,

або

– вузол типу *T*, з яким зв'язано кінцева кількість деревоподібних структур з базовим типом *T*, які звуться «піддеревами».

Варіант № 2 визначення. *Деревом* називають такий граф, у якого (рис. 2.18):

– тільки одна вершина, яку називають *коренем дерева,* має напівступінь входу рівний нулю, тобто до неї не заходить жодного ребра;
– всі інші вершини мають напівступінь входу рівний одиниці.

Якщо з такого графа видалити вершину, то структура, що залишиться, називається *лісом*.

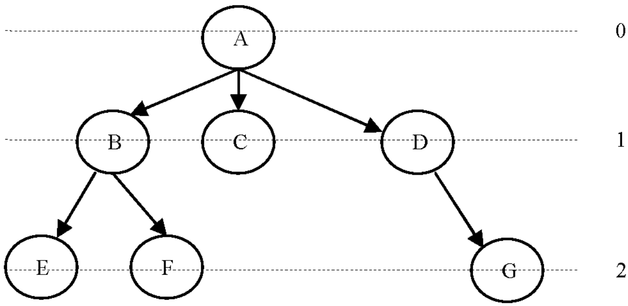


Рис. 2.18. Приклад дерева

Вершини дерева, які мають напівступінь виходу рівний нулю (з них не виходить жодного ребра), називають *листям дерева.* Всі інші вершини називають *вузлами дерева.* У дерева, що представлено на рис. 2.18, коренем є вершина A, листям – вершини E, F і G, а вузлами – вершини B, C і D. Якщо у вузла видалити ребро, що входить до нього, то отримаємо теж дерево, яке є піддеревом дерева. Ці піддерева називають ще *гілками*.

Наведене вище визначення дерева виходить з того, що цей спеціальний граф є орієнтованим (від вершини до листів), але на рисунках, як правило, орієнтацію не наводять, що буде використовуватися нижче.

Дерево називають m-арним, якщо хоча б один вузол має m ребер, що виходять із нього. Дерево, що представлено на рис. 1, є 3-нарним деревом. Якщо $m=2$, то дерево називають *бінарним*. Якщо корінь дерева і всі вузли дерева мають по два ребра, що виходять, то таке дерево називають *повністю бінарним*. За аналогією визначається повністю m-арне дерево. Будь яке m-арне дерево можна перетворити у бінарне і навпаки.

На рис. 2.19 наведено приклад бінарного дерева.

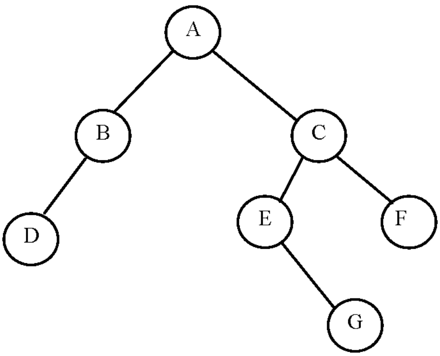


Рис. 2.19. Приклад бінарного дерева

У бінарного дерева для кореня і кожного вузла кожне із двох можливих ребер є або лівим (воно створює ліве піддерево), або правим (воно створює ліве піддерево). Так, на рис. 2.19 вершина B має тільки ліве піддерево, а вершина E має тільки праве піддерево.

Бинарні дерева є дуже розповсюдженою структурою даних. З їх використанням побудовано багато алгоритмів. Нижче розглянемо алгоритми обходу бінарних дерев.

Існує три способи (алгоритми) обходу бінарного дерева:

1) низхідний (спадний) або «згори донизу»;

2) висхідний (підіймальний) або «знизу догори»;

3) змішаний.

Кожен із способів має дві модифікації: «зліва направо» і «справа наліво». Якщо вершині дерева надати мнемоніку *V*, обходу «зліва направо» – *L*, а обходу «справа наліво» – *R*, то за їх допомогою можна позначити всі шість можливих способів обходу, при цьому має значення порядок розташування прийнятих позначень. Так, позначення *LRV* означає висхідний обхід (позначення вершини стоїть наприкінці) «зліва направо», а позначення *RVL* означає змішаний обхід (позначення вершини стоїть усередині) «справа наліво».

Алгоритми названих методів обходу наведені у таблиці 2.12. Вони є рекурсивними.

Таблиця 2.12

Алгоритми обходу бінарних дерев

Позначення	**Крок 1-й**	**Крок 2-й**	**Крок 3-й**
VLR	Обходимо вершину	Обходимо ліве піддерево	Обходимо праве піддерево
VRL	Обходимо вершину	Обходимо праве піддерево	Обходимо ліве піддерево
LRV	Обходимо ліве піддерево	Обходимо праве піддерево	Обходимо вершину
RLV		Обходимо ліве піддерево	Обходимо вершину
LVR	Обходимо ліве піддерево	Обходимо вершину	Обходимо праве піддерево
RVL	Обходимо праве піддерево	Обходимо вершину	Обходимо ліве піддерево

Хай задане бінарне дерево у табличній формі (див. табл. 2.13), де V_i – вершина, з якої виходить дуга, а V_j – вершина, до якої входить дуга.

Таблиця 2.13

Приклад бінарного дерева у табличному вигляді

V_i	a	a	b	b	c	c	d	d	e	e	f	f
V_j	b	c	d	e		f		g	h		i	j

Графічно це дерево представлено на рис. 2.20, а у таблиці 2.14 представлено результати обходу цього дерева всіма способами.

Таблиця 2.14

Результати обходу дерева, представленого на рис. 2.17, різними способами

Алгоритм обходу	VLR	VRL	LRV	RLV	LVR	RVL
Результат обходу	abdgehcfij	acfjibehdg	gdhebijfca	jifchegdba	dgbheacifj	jficaehbgd

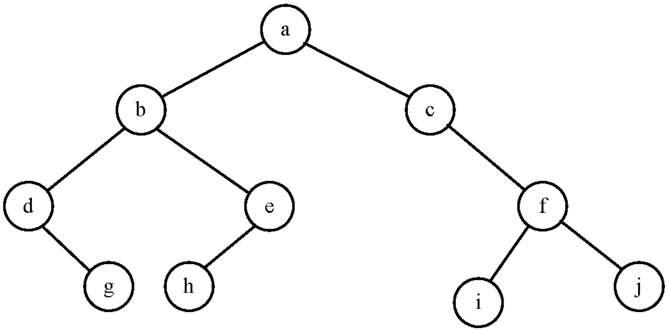
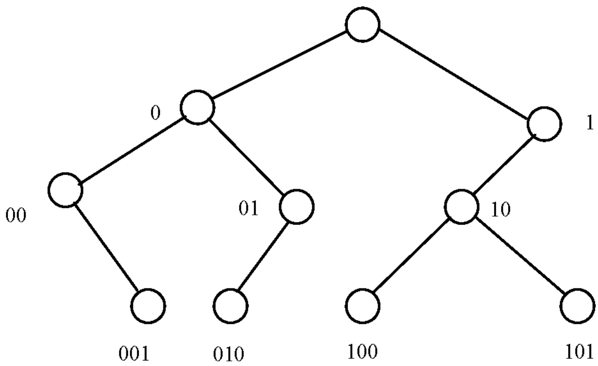


Рис. 2.20. Бінарне дерево, що задане у табл. 2.13

Крім табличної форми завдання дерев, бінарні дерева можна представити переліком кодів листів цього дерева, якщо прийняти таку систему кодування вершин:

- коди вершин складаються із символів *«0»* і *«1»;*
- коди вершин дерева мають і символів, де *i* – рівень (глибина) дерева, при цьому корінь дерева має нульовий рівень, тобто корінь дерева немає коду;
- код вершини створюється операцією конкатенації коду предка та символу *«0»* для лівого піддерева і *«1»* для правого піддерева.

На рис. 2.21 наведено приклад бінарного дерева з кодами за наведеними правилами і перелік кодів листів цього дерева, за яким можна відновити це графічне відображення дерева.



Перелік кодів листів (001, 010, 100, 101), що визначають дерево

Рис. 2.21. Приклад кодування вершин бінарного дерева

Оскільки для бінарних дерев вже розроблено й програмно реалізовано чимало алгоритмів, то інколи варто перетворити m-арне дерево в бінарне. Один із алгоритмів такого перетворення покажемо на прикладі m-арного дерева, що наведено на рис. 2.22.

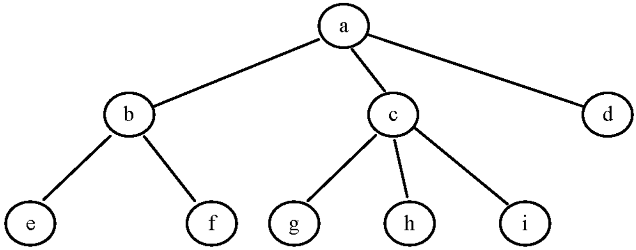


Рис. 2.22. Приклад m-арного дерева

Алгоритм перетворення m-арного дерева в бінарне без верхнього правого піддерева полягає у наступному. Починаючи з кореневої вершини дерева дугу крайнього лівого піддерева робимо вертикальною, такою що з'єднує старшу за ієрархією вершину з підпорядкованою, від якої послідовно зліва направо горизонтальними дугами з'єднуємо одну за одною вершини цього ж рівня (вершини-брати), потім розглядаємо кожну з цих вершин як кореневу і виконуємо з кожною з них вище наведені дії доки всі вершини, що розглядаються, не будуть листям. Всі вертикальні дуги утворюють ліві піддерева, а горизонтальні – праві. На рис. 2.23 зображено таке перетворення бінарного дерева, що на рис. 2.22, при цьому: зліва – за наведеним вище алгоритмом, справа – теж саме перетворене дерево у більш звичному представленні.

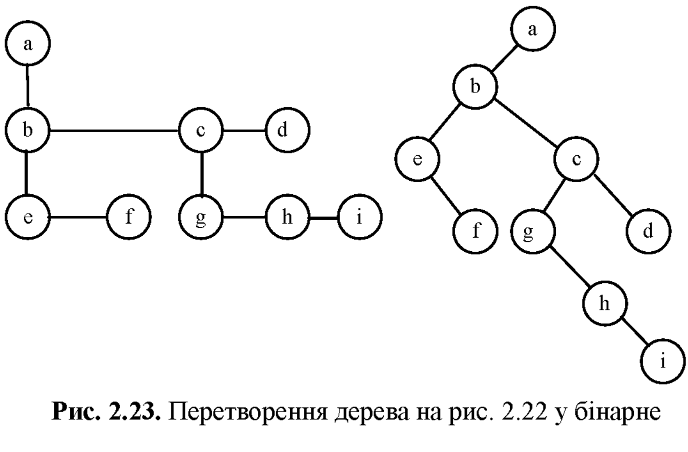


Рис. 2.23. Перетворення дерева на рис. 2.22 у бінарне