

ЛЕКЦІЯ 11.

НЕЛІНІЙНІ СТРУКТУРИ ДАНИХ

11.1. Спискові структури

Список, що містить всі підсписки, які відходять від нього, називають списковою структурою. Ці структури являють собою узагальнення лінійних списків.

Спискові структури описуються трьома характеристиками.

1. Порядок. Над елементами списку задано транзитивне відношення, що визначається послідовністю, в якій елементи з'являються в середині списку. Наприклад, у списку (x, y, z) елемент x передуює y і y передуює z . Список (x, y, z) не еквівалентний списку (y, z, x) . При графічному зображенні списків порядок визначається горизонтальними стрілками. При цьому розуміється, що елемент, з якого виходить стрілка, передуює елементу, в який заходить стрілка.

2. Глибина. Це максимальний рівень, який приписується елементам в середині списку, або в середині всякого підсписку в списку. Рівень елемента визначається в залежності від кількості підсписків, що містяться в середині списку, тобто числом пар дужок, якими оточується елемент.

Наприклад, на рис. 11.1 графічно зображений список $(a, (b, c, d), e, (f, g))$ у якого рівень елементів a і e дорівнює одиниці, а елементів b, c, d, f, g - двом. Глибина цього списку дорівнює двом.

При графічному зображенні списку поняття глибини і рівня полегшуються для розуміння, якщо кожному одиночному елементу приписати деяке число k . Значення k для елемента x , тобто $k(x)$, дорівнює кількості вертикальних стрілок, які необхідно пройти для того, щоб досягти даного з першого елемента списку. Наприклад, $k(a)=0$, $k(b)=1$ (рис. 11.1). Тоді рівень елемента задається відношенням $k(x)+1$, а глибина списку дорівнює максимальному значенню рівнів серед усіх рівнів елементів списку.

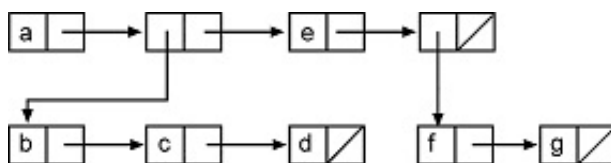


Рис.11.1. Спискова структура

Наприклад, максимальний рівень елементів у списку на рис.8.6 дорівнює двом, тому глибина списку також дорівнює двом.

3. Довжина. Це число елементів списку першого рівня.

Наприклад, довжина списку $(a, (b, c), d)$ дорівнює трьом, а списку, зображеного на рис.11.1 - чотирьом, тому що він містить чотири елементи: елемент a , підсписок (b, c, d) , елемент e і підсписок (f, g) .

11.1.1. Ієрархічні списки

Спискові структури можуть бути достатньо складними. Прикладом складної спискової структури можуть служити ієрархічні списки, які об'єднують у один список записи з різним

внутрішнім порядком. Така ієрархія записів може бути східчастою і достатньо глибокою. Якщо елементами списку верхнього рівня є підлеглі списки, то зручно вважати, що з точки зору верхнього рівня запис, що в нього входить, є посиланням на підлеглий список, оформлений як заголовок даного підлеглого списку. При цьому в тілі заголовка підписку може бути і деяка змістовна інформація загального характеру. Список верхнього рівня може містити поруч із заголовками підлеглих списків також самостійні елементи, тобто звичайні записи. Таким чином, список верхнього рівня зображується у вигляді ланцюга, в якого кожна ділянка складається з довідки і звичайного запису або заголовку підписку - списку з нижнього рівня ієрархії (рис. 11.2).

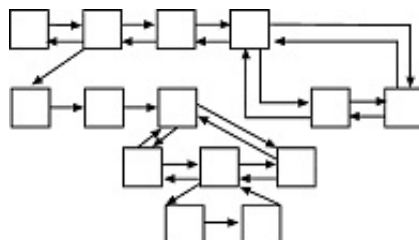


Рис.11.2. Приклад структури ієрархічного списку

На рис.11.2 ієрархічно пов'язані між собою списки різних типів. Двонаправлений список найвищого рівня складається з чотирьох елементів, причому перший і третій елементи є звичайними записами; другий елемент є однонаправленим списком, в якого третій елемент - двонаправлений кільцевий список, що, в свою чергу, складається зі звичайного запису, однонаправленого кільцевого списку і ще одного звичайного запису; четвертий елемент списку найвищого рівня являє собою двонаправлений кільцевий список із двох записів.

11.1.2. Організація спискових структур

Розглянемо зображення в пам'яті спискових структур. Очевидно, що для них найзручнішим є зчеплене зображення, яке при необхідності забезпечує динамічне розміщення вершин, легкість обробки і здатність розділяти підсписки. Список в загальному випадку задається деякою варіацією бінарного зображення натуральних дерев, шляхом використання двох полів зв'язку: одного - для вказівки членства в середині списку і другого - для вказівки адреси.

Типова структура елемента списку в машинному зображенні буде такою:

ПС ІЕ НС

Тут поле *ПС* вказує на перший елемент у підписку; поле *НС* - на наступний елемент у даному списку, а поле *ІЕ* містить інформацію безпосередньо про елемент даних.

Але такий формат елемента зручний тільки тоді, коли інформація про нього займає невеликий обсяг пам'яті, тоді поле *ІЕ* буде невеликим.

У більш загальному випадку для елементів даних потрібен відносно великий обсяг пам'яті, тому частіше використовуються формати, що складаються тільки з двох полів:

ПС	НС
----	----

Тут поле *НС* виконує ті самі функції, що й раніше, а поле *ПС* вказує або на перший елемент підписку, що входить у вузол списку, або на інформацію, пов'язану з ізольованим елементом списку, який утворює ізольований вузол. Припускається, що ізольовані вузли містять якусь унікальну характеристику, наприклад, поле маркера, що дозволяє відрізнити ізольовані вузли від спискових. Такий формат дозволяє зробити всі вузли в середині спискової структури однаковими за розміром і дає можливість уникнути зайвих втрат інформаційного простору у вузлах. У наведеному далі прикладі (рис. 11.3) елементи списку також зображені у форматі з двох полів, де у полі *ПС* означає вказівник на інформацію, пов'язану з ізольованим елементом списку.

Приклади. Нехай маємо спискові структури в дужковому зображенні, де в дужки взяті списки, а елементи розділяються комами (рис. 11.3):

а) $(a, (b, c, d), e, (f, g))$ - довжина чотири (два ізольованих елементи списку - a, e і два підписки - $(b, c, d), (f, g)$).

б) $()$ - нуль-список, довжина нуль;

в) $((a))$ - довжина одиниця (ізольований елемент списку утворює підписок).

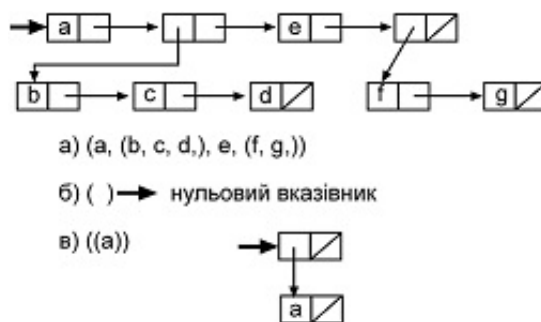


Рис. 11.3. Зображення спискових структур у пам'яті

Між списковими структурами і орієнтованими графами існує прямий зв'язок. Зокрема, список являє собою орієнтований граф з однією початковою вершиною, що відповідає входу в список. Кожна вершина безпосередньо пов'язана з початковою вершиною, що відповідає елементу списку: або з вершиною з півступінню виходу нуль (для ізольованих), або з вершиною, що має вихідні гілки (для елементів, які самі є списками). Ребра, що виходять із вершин, можна розглядати як впорядковані списки. Це означає, що розпізнається перше ребро, друге і т.д., які відповідають впорядкованим списковим елементам для першого, другого і т.д. елементів.

Тоді списки, зображені на рис. 11.3, можна змалювати також у вигляді графів (рис. 11.4).

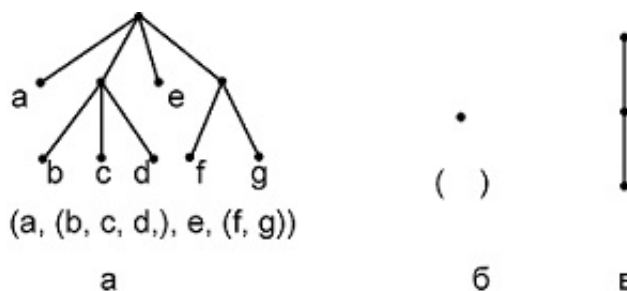


Рис. 11.4. Зображення спискових структур у вигляді графів

Існують списки, які не можуть бути зображені як дерева, але кожне дерево може бути зображене у вигляді якогось списку. Списки можуть мати внутрішню рекурсивну структуру, чого не мають дерева. Такі списки відповідають безкінечним графам.

Отже, для зображення в пам'яті самих спискових структур існує багато способів, але всі вони мають спільні риси, які опираються на використання бінарних дерев для зображення лісів. При цьому одне з полів у записі вузла використовується для вказівника на наступний елемент списку, а друге поле можна використати для вказівника на перший елемент підсписку.

11.2. Сіткові структури

Якщо у відношенні між даними породжений елемент має більше, ніж один вхідний або породжуючий елемент, то таке відношення уже не можна описати як деревоподібну або ієрархічну структуру, його описують у вигляді сіткової структура. Отже, **сіткова структура** це орієнтований зв'язний граф без циклів, у якого будь-який породжений вузол може мати більше одного породжуючого вузла і, крім того, всі вузли розміщуються так, що породжені елементи розміщуються нижче породжуючих. У сітковій структурі будь-який елемент може бути пов'язаний з будь-яким іншим елементом (рис.11.5).

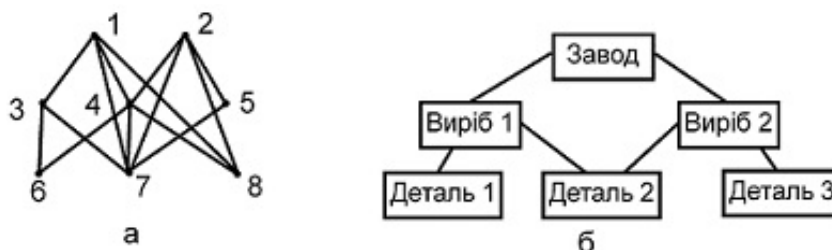


Рис.11.5. Елементи сіткових структур:

а) однорідні; б) неоднорідні

Однорідна сіткова структура, що містить однотипні елементи, вироджується в дерево або список.

Неоднорідна сіткова структура, в якій зображення відношення батько-син є складним (тобто в одного батька може бути декілька синів), а зображення відношення син-батько простим (у сина може бути тільки один батько), переважно також зводиться до дерева. Сіткову структуру з таким зображенням відношень називають простою.

Якщо зображення відношень син-батько також складне, таку структуру називають **складною**. На рис.11.5 зображено складні сіткові структури. Існує скорочений спосіб графічного зображення складних відношень - здвоєні стрілки (рис.11.6). Прості відношення зображуються однією стрілкою і називаються зв'язком типу $I:M$. Відношення між записами, що відображають зв'язки зі здвоєними стрілками в обох напрямках, прийнято називати зв'язком типу $M:N$. Майже завжди у складних сіткових структурах при зв'язках типу $M:N$ виникають дані перетину, які породжуються з різнотипності вузлів сітки. Наприклад, у відношенні постачальник-виріб такими даними перетину є ціна.

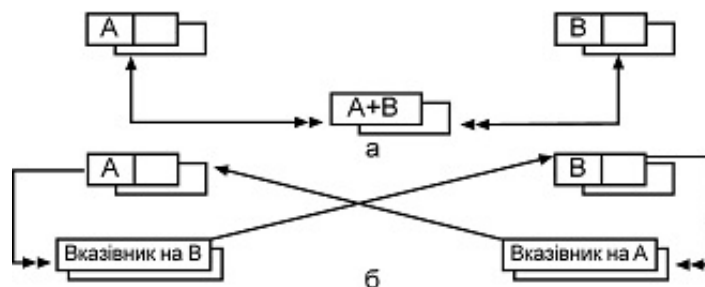


Рис. 11.6. Приклади зображення даних перетину:

а) за допомогою файлу; б) за допомогою вказівників

В основному процесі експлуатації бази даних виникають дві складні задачі:

- 1) зображення даних перетину у сіткових структурах;
- 2) доповнення даних перетину у базу даних.

На рис. 11.6 показано два способи підтримки зв'язків типу $M:N$ між записами з ключами A і B з метою фіксації даних перетину: на рис. 11.6,а є файл записів A та файл записів B і окремо інформація про зв'язки між цими файлами; на рис.11.6,б використовується ланцюг вказівників, який з'єднує кожний запис файлу A з відповідними йому записами файлу B . Введення в структуру даних перетину фактично руйнує зв'язок типу $M:N$.

Деякі сіткові структури містять цикли. **Циклом** у такій схемі вважається ситуація, коли попередник вузла в той же час є його послідовником. Відношення вхідний-породжений утворюють при цьому замкнутий контур. Наприклад, завод випускав різну продукцію. Деякі вироби виготовляються на інших заводах-субпідрядниках. З одним договором може бути пов'язано виготовлення декількох виробів. Зображення цих відношень утворює цикл (рис.11.7,а).

Деколи записи файлу зв'язані з іншими записами цього ж файлу. Таку ситуацію називають **петлею**. В петлі міститься тільки один тип запису, тобто тип породженого запису зливається з типом вхідного (рис.11.7,б).

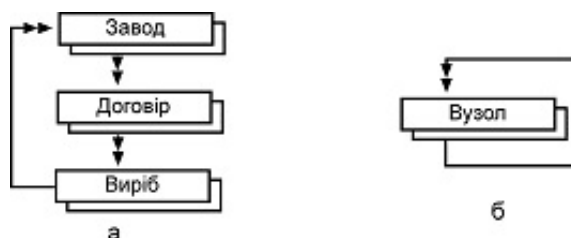


Рис.11.7. Цикли і петлі в сітках: а) цикл; б) петля

Вся сіткова структура може бути зведена до більш простого вигляду введенням надлишковості. Причому така надлишковість відноситься не до всього запису, а тільки до первинного ключа або способу його ідентифікації. Таким же чином перетворюються до деревоподібних структур і складні сітки - введенням надлишковості в ключах (рис.11.8).

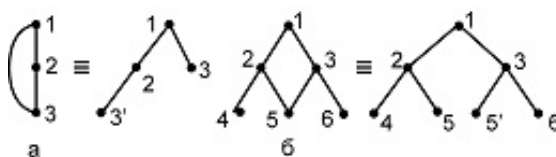


Рис.11.8. Перетворення сіток до дерев:

а) звичайні сітки; б) решітки

Перетворення сітки до дерева здійснюється дублюванням вершин, в які заходить декілька ребер. Очевидно, що кількість повторень і рівень таких вершин у дереві залежить від кількості її вхідних ребер і вершин у сітці. Наприклад, на рис 11.8,а вершина 3 записана у дереві двічі на другому і третьому рівнях, тому що в початковій сітці в неї заходить два ребра - з вершин 1 і 2.

Решіткою називають частковий випадок сіткової структури, в якій граф є кореневим і вхідні вузли деякого породженого вузла належать одному і тому ж рівню. На рис.11.8,б зображено решітку. Всі ієрархічні структури, які мають спільні підструктури, являють собою решітки.