

Системне програмування I

*Пустоваров В. І., НТУУ"КПІ",
м. Київ pust@vipustovarov@ukr.net*

Лекція 1/13

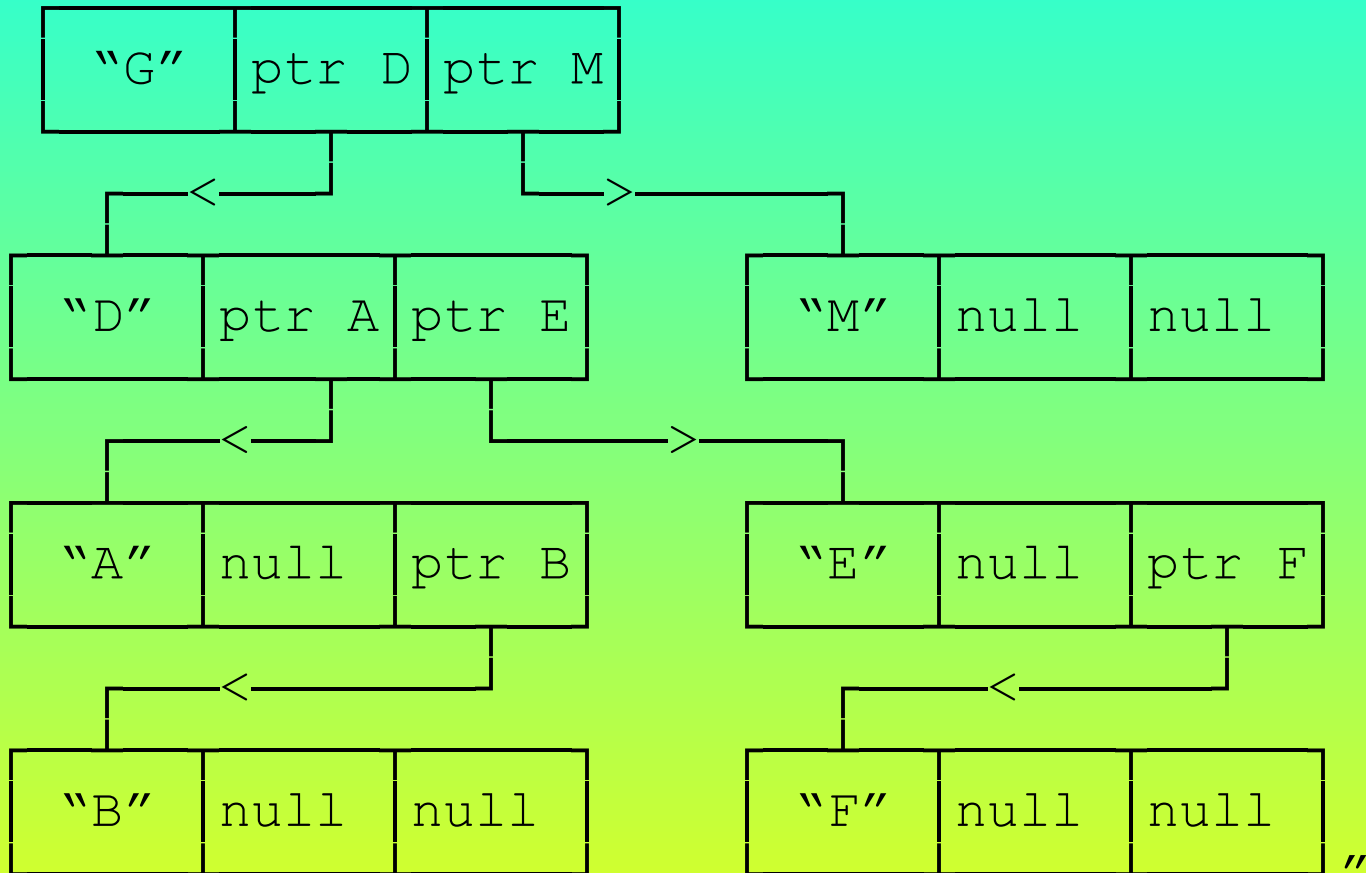
Організація таблиць та індексів впорядкування у вигляді масивів та структур з посиланнями

- 1. Деревоподібні таблиці та індекси.*
- 2. Організація пошуку за мірами близькості*
- 3. Модульний контроль*

Опис прикладу бінарного дерева

- Найпростіша реалізація цього методу базується на впорядкуванні елементів через бінарне дерево, в якому до кожного вузла може бути "підвішено" не більше двох піддерев. Кожний вузол дерева являє собою сформований елемент таблиці, причому кореневий вузол є першим елементом. Перші за порядком вказівники посилаються на елементи з меншим аргументом, а другі – на елементи з більшим аргументом. Припустимо тепер, що до таблиці необхідно записати ім'я "D". Для нього обирається ліве піддерево, тому що "D" < "G" знаходиться в корені.
- Тепер запишемо ім'я "M". Тому що "G" < "N", для збереження "M" обирається праве піддерево вузла, що зберігає ім'я "G". І, насамкінець, запишемо ім'я "E". Тому що "E" < "G", йдемо за лівою дугою і потрапляємо на ім'я "D". Оскільки "D" < "E", то обираємо дугу, що веде вправо від "D".
- Принцип побудови бінарного дерева такий, що для дерев з однаковим вмістом можна побудувати багато різних варіантів, які відрізняються коренями та проміжними вершинами. Кращі характеристики мають збалансовані дерева, тобто такі, в яких кількість рівнів зв'язків в ієрархії елементів не відрізняється за будь-якою парою шляхів більш ніж на 1.
- Формальну специфікацію обмежень пошуку в бінарних деревах можна визначити через відношення порядку елементів сусідніх рівнів:
- $A_{i+1,2j-1} < A_{ij} < A_{i+1,2j}$,
- де i – номер рівня елементів, j – номер елемента на відповідному рівні.

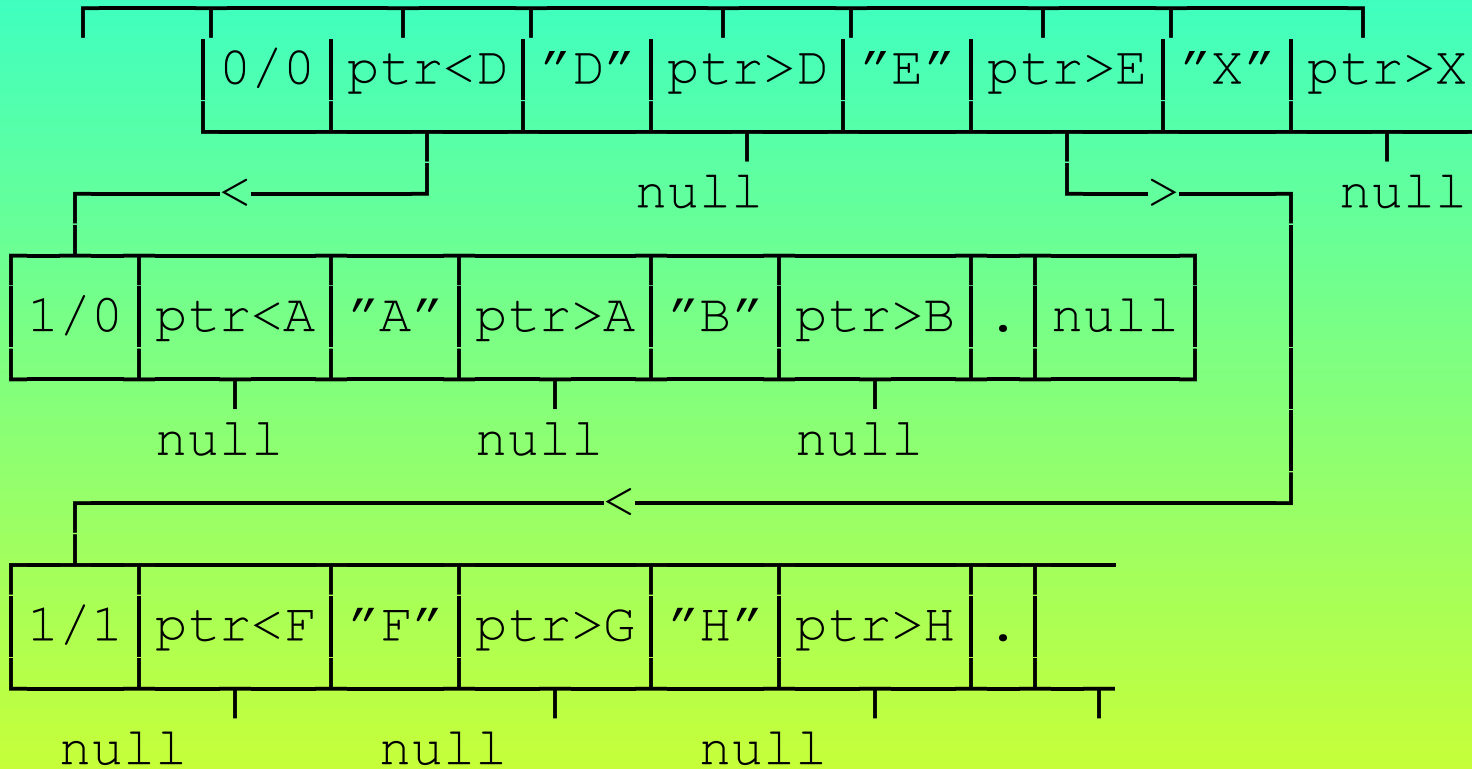
Приклад бінарного дерева



Опис прикладу В-дерева

- Подальший розвиток методів роботи з деревоподібними структурами призвів до використання більш розгалужених деревоподібних структур, що одержали назву В-дерев. В них зберігаються відношення впорядкування зв'язків вузлів і додаються відношення впорядкування внутрішніх елементів одного вузла, які мають бути узгоджені. Для лексикографічних порівнянь це однакові відношення аргументів попередніх та наступних елементів. Вузли В-дерева зберігають інформацію про декілька впорядкованих елементів. Звичайно кількість елементів у вузлі невелика і лежить в межах $3 < n < 256$.
- Як закони впорядкування даних у вузлах В-дерев та між їхніми вузлами можуть використовуватись правила впорядкування за алфавітом та правила впорядкування за хеш-функціями. На наступному слайді наведено приклад В-дерева, що вміщує, ту ж інформацію, що і бінарне дерево з рис. 13.1, але займає лише два рівні ієрархії. При використанні хеш-функцій для впорядкування в вузлі дерева доцільні алгоритми заповнення вузлів з гори до низу, а при алфавітному впорядкуванні вузлів - побудова вузлів нових рівнів з низу до гори.

Приклад B-дерева



ПОБУДОВА ПРОЦЕДУР ВИЗНАЧЕННЯ МІР БЛИЗЬКОСТІ ДЛЯ КЛЮЧОВИХ ПОЛІВ

- При пошуку помилково підготовлених слів в текстових редакторах та процесорах часто виникає потреба в визначенні схожості ключів пошуку. Такі дії часто виконуються в текстовому процесорі MS Word. Вони можуть будуватися на підрахунку кількості однакових n_e , схожих літер n_{si} за i -м типом схожості, а також літер, які не мають відповідника в іншому ключі і можуть спиратися на абсолютні і відносні формульні критерії схожості. Схожість літер може визначатися залежно від випадку аналізу за схожістю написання літер в різних алфавітах n_{s1} , за близькістю комп'ютерних кодів n_{s2} та за близькістю розташування на клавіатурі n_{s3} , а також з урахуванням кількості літер n_{s4} , які не мають відповідників в обох ключах.
- При створенні програм порівняння за мірою близькості треба побудувати загальний критерій близькості як монотонну функцію $f(n_{s1}, n_{s2}, n_{s3}, n_{s4})$ в одному напрямку від n_{s1}, n_{s2} і n_{s3} та в іншому напрямку від n_{s4} . Крім того, попередньо необхідно організувати підрахунок n_{s1}, n_{s2}, n_{s3} і n_{s4} , при порівняльному перегляді ключів, які порівнюються. Результат пошуку за таким критерієм може бути неоднозначним, навіть за умови вимоги однозначності ключів. На алгоритм лінійного пошуку це практичного не впливає, а у випадку базового двійкового пошуку доцільно починати пошук навколо найближчого ключа, знайденого за відношенням порядку.

Підсумки

- Найбільш зручними методами організації роботи з впорядкованими таблицями є побудова таблиць та індексів з деревоподібними структурами.
- Двійкові дерева часто використовуються для побудови індексів в базах даних і системних програмах
- Розгалужені В-дерева використовуються для побудови комбінованих індексів в базах даних і системних програмах