TRƯỜNG ĐẠI HỌC AN GIANG
KHOA CÔNG NGHỆ THÔNG TIN

PHÂN TÍCH THIẾT KẾ GIẢI THUẬT

Giảng viên phụ trách:
NGUYỄN THÁI DƯ

Chương 5 B-TREE

- Giới thiệu
- Định nghĩa B-Tree
- Các phép toán trên B-Tree

Giới thiệu

- Cây 2-3-4 là một ví dụ về cây nhiều nhánh, trong cây nhiều nhánh mỗi node sẽ có nhiều hơn hai node con và nhiều hơn một mục dữ liệu.
- Một loại khác của cây nhiều nhánh là B-tree, là cây rất hiệu quả khi dữ liệu nằm trong bộ nhớ ngoài.

Định nghĩa B-Tree

- Một B-tree **bậc n** có các đặc tính sau:

  i) Mỗi node có tối đa 2*n khoá.

  ii) Mỗi node (không là node gốc) có ít nhất là n khoá.

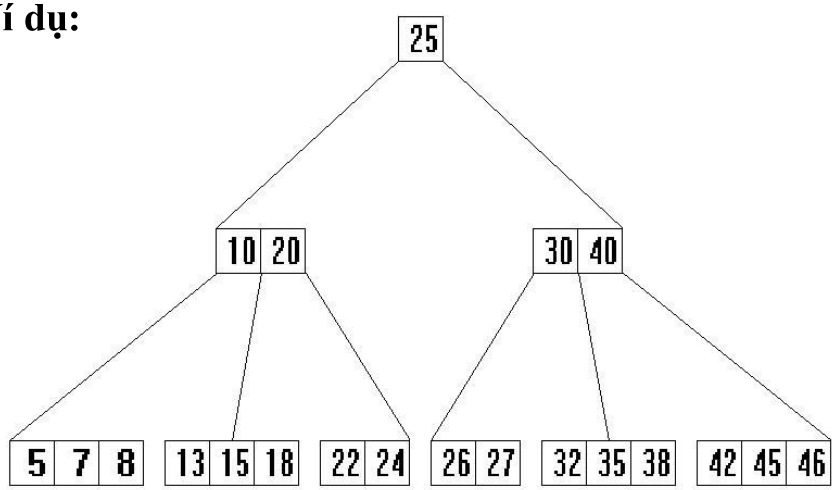
  iii) Mỗi node hoặc là node lá hoặc có m+1 node con (m là số khoá của trang này)

  iv) Các khóa được sắp tăng dần từ trái sang phải

  v) Các nút lá nằm cùng một mức

Định nghĩa B-Tree

Ví dụ:



B-tree bậc 2 có 3 mức

Ưu điểm B-Tree

- B-Tree là dạng cây cân bằng, phù hợp với việc lưu trữ trên đĩa
- B_Tree tiêu tốn số phép truy xuất đĩa tối thiểu cho các thao tác
- Có thể quản lý số phần tử rất lớn

Các phép toán trên B-Tree

- Tìm 1 phần tử có khóa bằng X trong cây
- Thêm 1 khoá vào vào B –Tree
- Xóa 1 khoá trong 1 nút

Tìm kiếm phần tử có khóa X trên cây

- Khoá cần tìm là X.Với m đủ lớn ta sử dụng phương pháp tìm kiếm nhị phân, nếu m nhỏ ta sử dụng phuong pháp tìm kiếm tuần tự. Nếu X không tìm thấy sẽ có 3 trường hợp sau xảy ra:

  i) $K_i < X < K_{i+1}$. Tiếp tục tìm kiếm trên cây con C_i

  ii) $K_m < X$. Tiếp tục tìm kiếm trên C_m

  iii) $X < K_1$. tiếp tục tìm kiếm trên C_0

Quá trình này tiếp tục cho đến khi node đúng được tìm thấy. Nếu đã đi đến node lá mà vẫn không tìm thấy khoá, việc tìm kiếm là thất bại.

Thêm 1 nút vào B-Tree

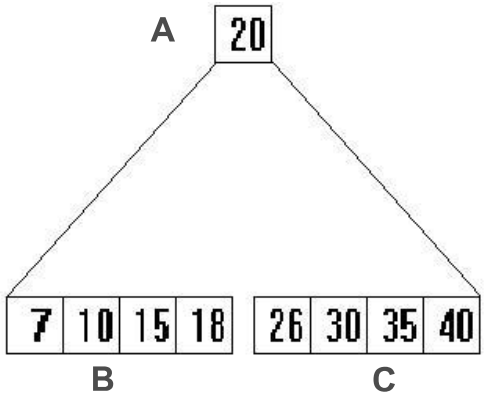
- Tính chất B-Tree: một node có ít nhất một nữa số khóa
- Thêm 1 nút có khóa X vào B-Tree
  - Thêm X vào 1 nút lá
  - Sau khi thêm, nếu nút lá đầy:
    - Tách nút lá ra làm đôi
    - Chuyển phần tử giữa lên nút cha và lan truyền ngược về gốc.
    - Nếu gốc bị tách, cây được đặt ở mức sâu hơn

Thêm 1 nút vào B-Tree

- Nếu số khóa lớn hơn 2n thì tách node:
  - Đưa phần tử giữa lên node cha
  - Tạo thêm node mới
  - Chuyển dời một nửa phần tử sang node mới
  - Tiếp tục lan truyền ở node cha (nếu node cha sau khi thêm > 2n phần tử thì thực hiện tách node như trên).

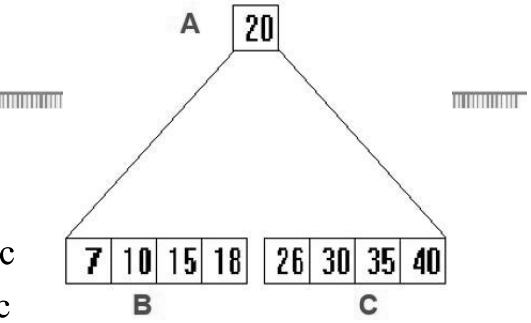
Thêm vào

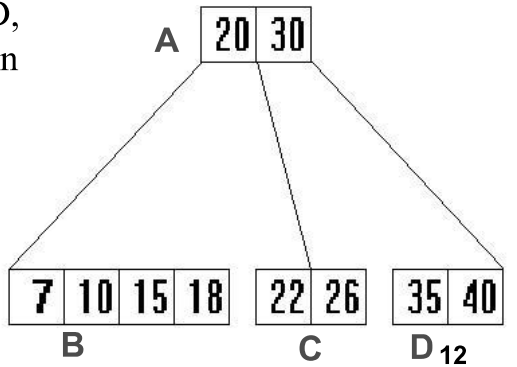
- **Ví dụ 1:**
  - Thêm x=22 vào B-Tree bậc 2. Khóa 22 chưa có trong cây. Nhưng không thể thêm vào node C vì node C đã đầy.



Thêm vào

Do đó tách node C thành hai node : node mới D được cấp phát và m+1 khóa được chia đều cho 2 node C và D, và khóa ở giữa được chuyển lên node cha A



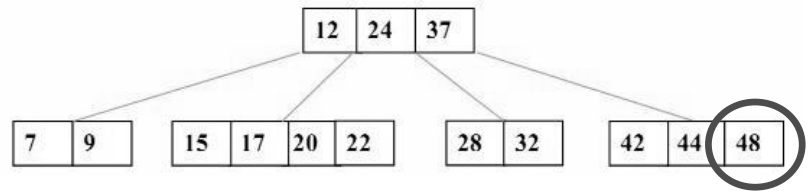


Xóa 1 phần tử trên B-Cây bậc n

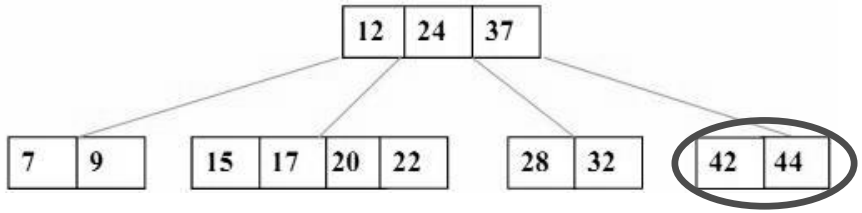
- Khóa cần xóa trên node lá -> Xóa bình thường.
- Khóa cần hủy không trên node lá:
  - Tìm phần tử thay thế: Trái nhất (hoặc phải nhất) trên hai cây con cần tìm
  - Thay thế cho nút cần xóa
- Sau khi xóa, node bị thiếu (vi phạm đk B-Tree):
  - Hoặc chuyển dời phần tử từ node thừa
  - Hoặc ghép với node bên cạnh (trái/phải)

Ví dụ về xóa

- Giả sử đã xây dựng B-Tree như sau:

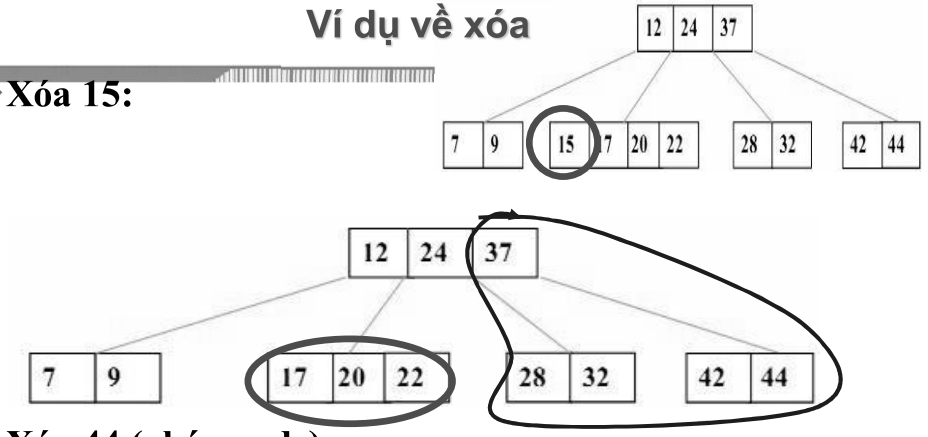


- **Xóa 48**

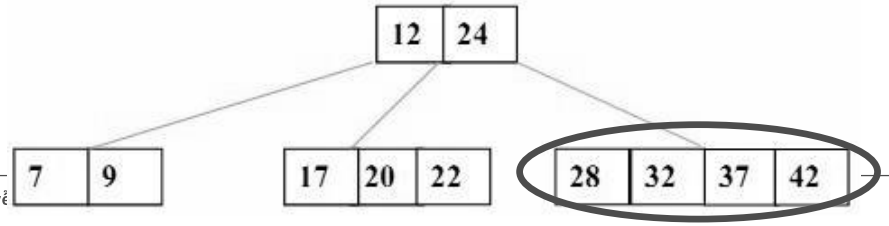


Ví dụ về xóa

- **Xóa 15:**

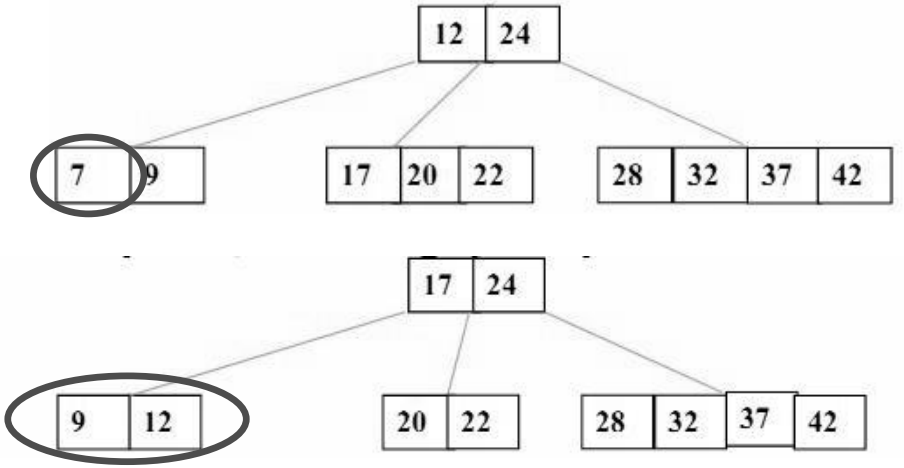


- **Xóa 44 (ghép node)**

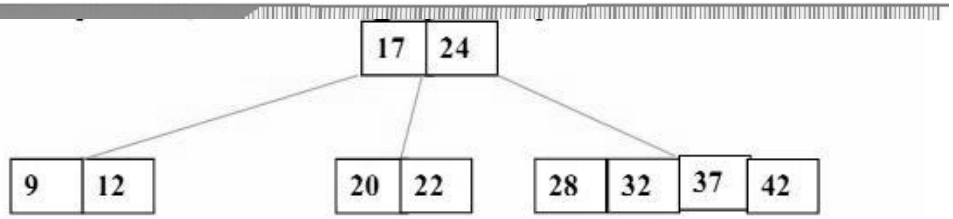


Ví dụ về xóa

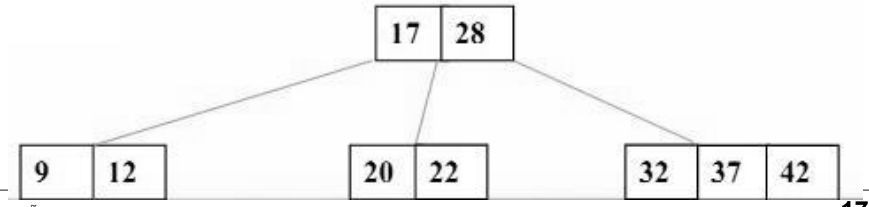
- **Xóa 7 (mượn node phải):**



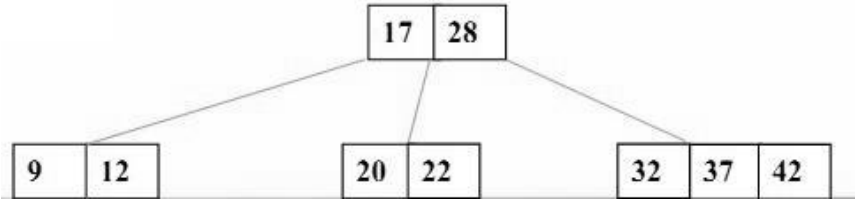
Ví dụ về xóa



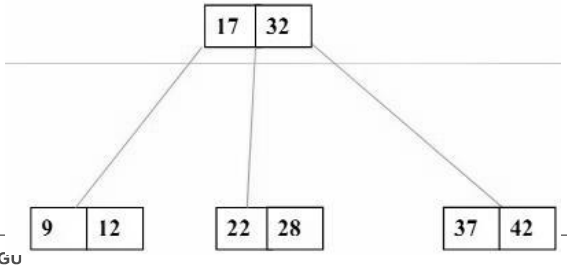
◆**Xóa 24**, ta đem khóa 22 lên thay thế. Khi đó node 20,22 chỉ còn 20 (phạm), ta phải đem khóa 22 trở lại node 20,22. Mang 28 lên thêm



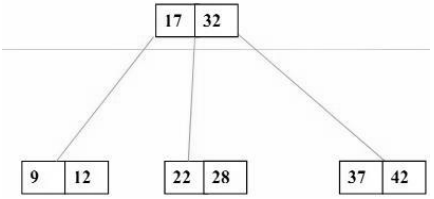
Ví dụ về xóa



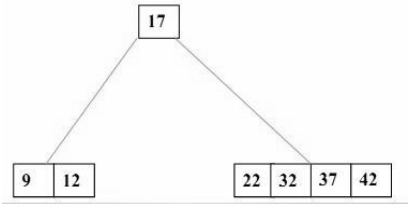
◆**Xóa 20**: Mượn node phải 1 phần tử. Tức mang 32 lên cha, 28 xuống node có 1 khóa là 22



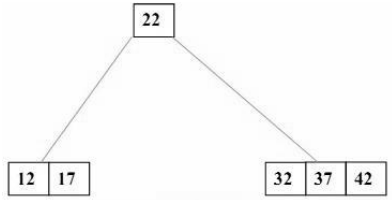
Ví dụ về xóa



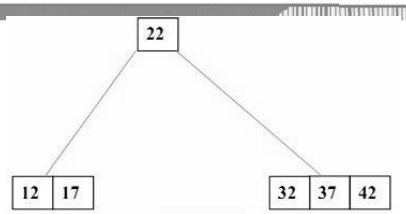
◆**Xóa 28:**



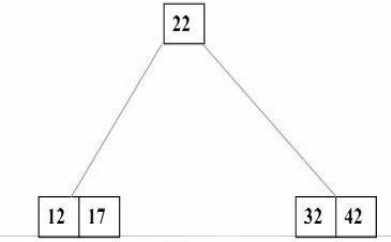
◆**Xóa 9:**



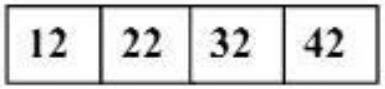
Ví dụ về xóa



◆**Xóa 37:**



◆**Xóa 17:**

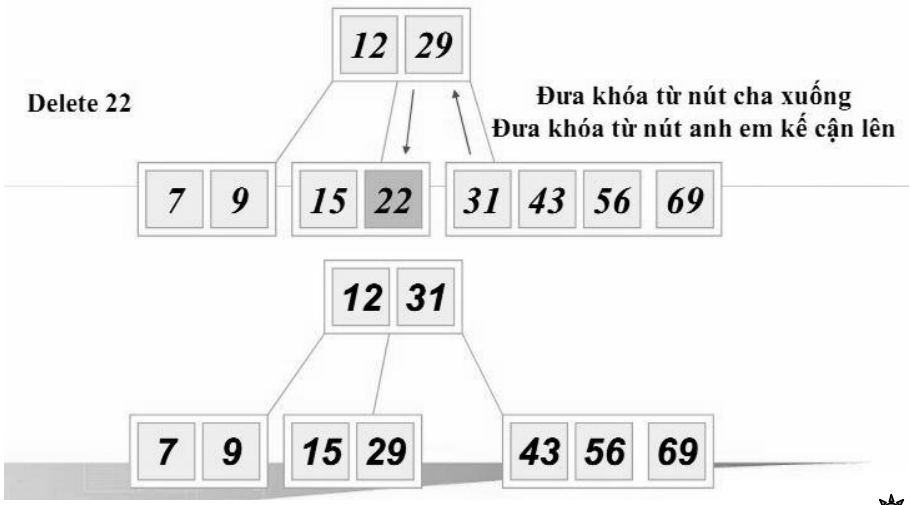


B-tree: Cân bằng lại cây sau khi xóa

- Nếu một trong các nút anh em kế cận nút đang xét có số lượng khóa nhiều hơn số lượng tối thiểu
  - Đưa một khóa của nút anh em lên nút cha.
  - Đưa một khóa ở nút cha xuống nút đang xét.
- Nếu tất cả các nút anh em kế cận nút đang xét đều có số lượng khóa vừa đủ số lượng tối thiểu
  - Chọn một nút anh em kế cận và hợp nhất nút anh em này với nút đang xét và với khóa tương ứng ở nút cha.
  - Nếu nút cha trở nên thiếu khóa, lặp lại quá trình này.

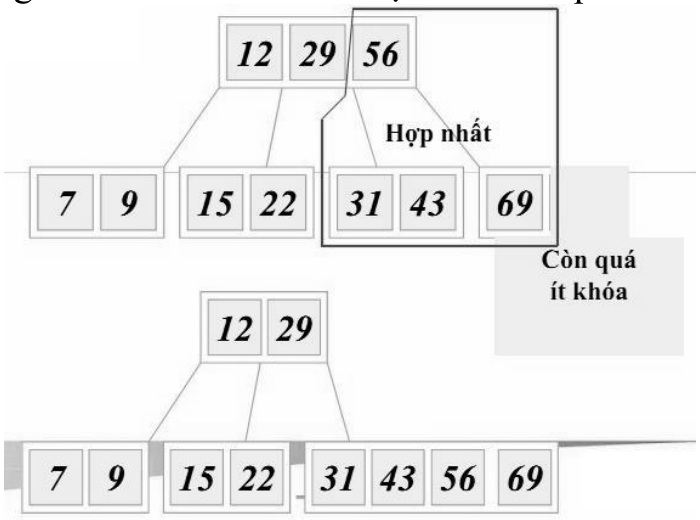
Trường hợp:

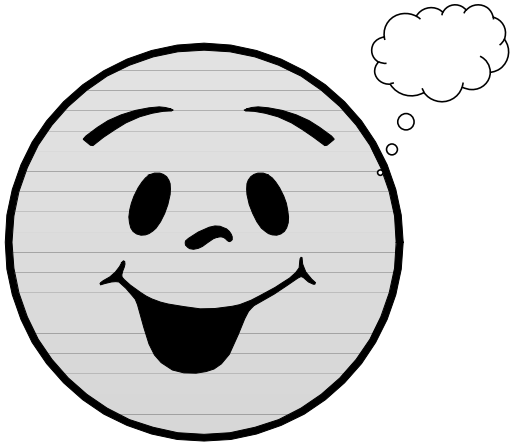
- Nút anh em kế cận còn đủ khóa để bổ sung



Trường hợp:

- Nút đang xét và nút anh em kế cận đều còn quá ít khóa





Cảm ơn !