

ĐHSP

## B - cây



<http://fit.hcmup.edu.vn/~tien>



## Nội dung

1

Đặt vấn đề

2

B-cây: Định nghĩa

3

Thêm phần tử

4

Xóa phần tử

[www.themegallery.com](http://www.themegallery.com)





## Đặt vấn đề

- ❖ Cần lưu trữ số phần tử dữ liệu rất lớn
- ❖ Lưu trữ trên bộ nhớ ngoài
- ❖ Tìm kiếm nhanh

[www.themegallery.com](http://www.themegallery.com)



## Giải quyết

- ❖ **Thao tác tìm kiếm trên cây tỉ lệ thuận với chiều cao của cây. Nếu cây có N nút thì.**
  - Chiều cao của cây nhị phân tìm kiếm cân bằng  $\leq \log_2 N =$  chi phí tìm kiếm.
  - Cây 2-3-4 (mỗi nút có nhiều nhất 4 khoá)  $\rightarrow \leq$  chiều cao  $\log_4 N$ .
  - Chưa phù hợp khi lưu trữ và truy xuất trên đĩa

[www.themegallery.com](http://www.themegallery.com)



## Giải quyết

- ❖ Dùng 1 loại cây khác nhằm mục đích
  - Phân trang dữ liệu
    - Tăng số nhánh của cây → Giảm chiều cao của cây
    - Gom nhóm dữ liệu thành những block → giảm số lần truy xuất trên đĩa
- ⇒ **B-Cây** (1 loại cây nhiều nhánh) thích hợp với việc lưu trữ và truy xuất trên bộ nhớ ngoại - đĩa cứng

www.themegallery.com



## B-cây: Định nghĩa

B-Tree bậc  $m$  là cây có các tính chất:

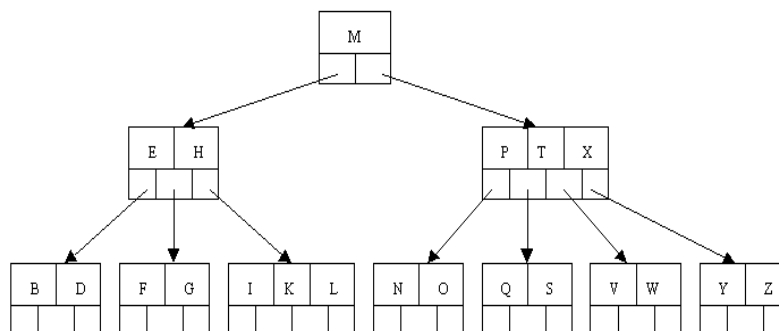
1. Mỗi nút có tối đa  $m$  nút con.
2. Mỗi node (ngoại trừ gốc) có ít nhất  $\lceil m/2 \rceil$  nút con.
3. Nút gốc (nếu không phải nút lá) có ít nhất 2 nút con.
4. Mọi nút lá đều nằm cùng một mức.
5. Mọi nút khác lá có  $k$  nút con sẽ có  $k-1$  khóa.
6. Các khóa và cây con được sắp xếp theo cây tìm kiếm.

www.themegallery.com



## B-cây: Ví dụ

❖ B-cây bậc 5 có 21 phần tử



## B-cây: Ưu điểm

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



## B-cây: Các phép toán trên cây

- ❖ Thêm một khóa vào B-cây.
- ❖ Xóa 1 khóa trong một nút.
- ❖ Tìm một phần tử có khóa bằng x trong cây.

[www.themegallery.com](http://www.themegallery.com)



## B-cây: Tìm kiếm

- ❖ Việc tìm kiếm được thực hiện tuần tự giống như trong cây nhị phân tìm kiếm:
  - Bắt đầu từ gốc.
  - Duyệt cây theo kiểu top-down.
  - Tại mỗi node so sánh khóa cần tìm với các giá trị khóa của node đó để tìm nhánh con cần duyệt tiếp theo.
- ❖ Có thể áp dụng phương pháp tìm kiếm nhị phân để tìm một khóa trong nội bộ một node.

[www.themegallery.com](http://www.themegallery.com)





## B-cây: Thêm một khóa mới

❖ **Ý tưởng:** Tìm vị trí khóa có thể thêm vào cây. Việc tìm kiếm sẽ kết thúc tại một lá. Khóa mới sẽ được thêm vào nút lá.

Chưa đầy → Việc thêm hoàn tất.

Đầy → Phân đôi nút lá cần thêm:

- Tách nút lá ra làm hai nút cạnh nhau trong cùng một mức.
- Chuyển phần tử giữa lên nút cha.

Quá trình phân đôi các nút có thể được lan truyền ngược về gốc và kết thúc khi có một nút cha nào đó cần được thêm một khóa gọi từ dưới lên mà chưa đầy.

[www.themegallery.com](http://www.themegallery.com)



## Thêm một khóa vào B-cây: Ví dụ

❖ Cho B-tree rỗng. Lần lượt thêm giá trị các khóa sau (theo thứ tự) vào B-tree: **1 12 8 2 25 5 14 28 17**

**7 52 16 48 68 3 26 29 53 55 45**

❖ Cần xây dựng B-tree bậc 5.

[www.themegallery.com](http://www.themegallery.com)

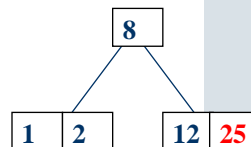


## Thêm một khóa vào B-cây: Ví dụ

❖ Bốn phần tử đầu tiên được đưa vào nút gốc

1	2	8	12
---	---	---	----

❖ Nếu đưa phần tử thứ năm vào nút gốc sẽ làm vi phạm điều kiện cây B-tree.



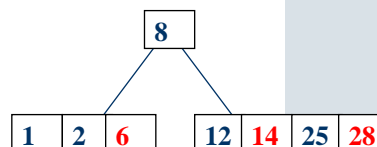
❖ Do đó, khi thêm khóa **25**, tách nút gốc thành 2 nút và đưa khóa ở giữa lên để tạo nút gốc mới.

www.themegallery.com

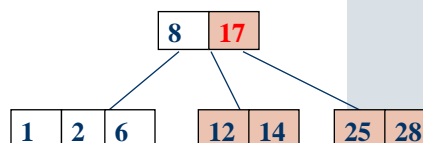


## Thêm một khóa vào B-cây: Ví dụ

Thêm các khóa **6, 14, 28** vào các nút lá:



Nút lá bên phải đã đầy (đã có 4 phần tử). Do đó, khi thêm khóa **17** vào nút lá bên phải sẽ làm nút lá bị “quá tải”. Lấy khóa ở giữa đưa lên nút cha (hiện tại là nút gốc) và phân chia nút hiện tại.

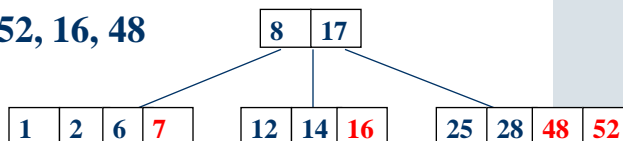


www.themegallery.com



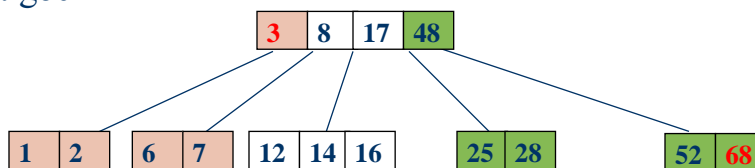
## Thêm một khóa vào B-cây: Ví dụ

Thêm các khóa **7, 52, 16, 48**  
vào các nút lá



Thêm khóa **68** vào, cần tách nút lá ở bên phải, đưa **48** lên nút gốc

Thêm khóa **3** vào, cần tách nút lá ở bên trái, đưa khóa **3** lên nút gốc

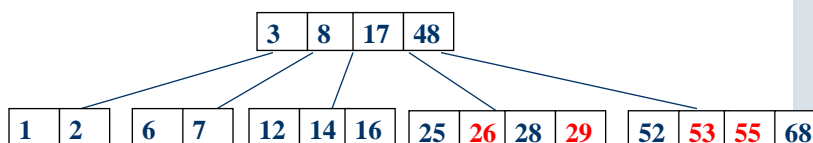


www.themegallery.com



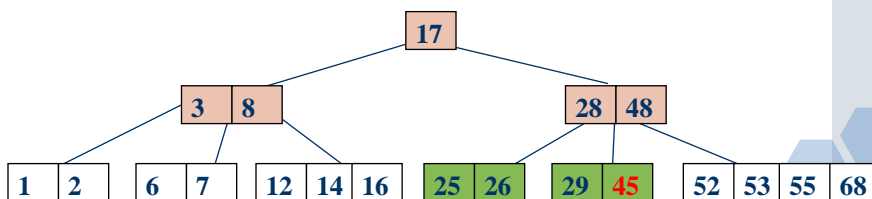
## Thêm một khóa vào B-cây: Ví dụ

Thêm các khóa **26, 29, 53, 55** vào các nút lá



Thêm khóa **45** vào, cần tách [25, 26, 28, 29] và đưa khóa **28** lên nút gốc.

Khi đó, nút gốc sẽ “quá tải” và cần tách tiếp



www.themegallery.com





## B-tree: Ví dụ áp dụng

Chèn các khóa sau (theo thứ tự) vào B-tree bậc 5:

❖ **3, 7, 9, 23, 45, 1, 5, 14, 25, 24, 13, 11, 8, 19, 4, 31, 35, 56**

❖ **C N G A H E K Q M F W L T Z D P R X  
Y S**

[www.themegallery.com](http://www.themegallery.com)



## B-tree: Xóa một khóa

❖ **Có 2 trường hợp cần xem xét khi xóa một khóa:**

- Khóa này thuộc node trong và là giá trị phân hoạch cho các node con.
- Xóa một khóa có thể làm cho số khóa và số con của nút hiện tại nhỏ hơn ngưỡng tối thiểu mà một node trong cây B-tree phải thỏa.

[www.themegallery.com](http://www.themegallery.com)



## B-tree: Xóa một khóa

- ❖ Nếu khóa này thuộc node lá, đơn giản xóa khóa này khỏi node lá đó.
- ❖ Nếu khóa này thuộc một node trong:
  - Tìm khóa lớn nhất thuộc cây con trái (phải nhất) hoặc khóa nhỏ nhất thuộc cây con phải (trái nhất) đưa lên thay thế cho khóa cần xóa.
  - Xóa khóa trái nhất hoặc phải nhất ở cây con tương ứng.

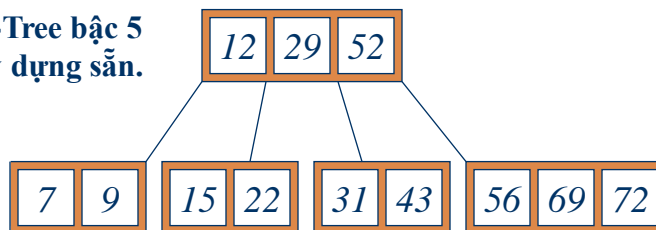
**Chú ý:** việc xóa 1 khóa khỏi một node có thể đòi hỏi phải cân bằng lại cây.

[www.themegallery.com](http://www.themegallery.com)



## Trường hợp: Xóa khóa ở nút lá

Xóa B-Tree bậc 5 đã xây dựng sẵn.

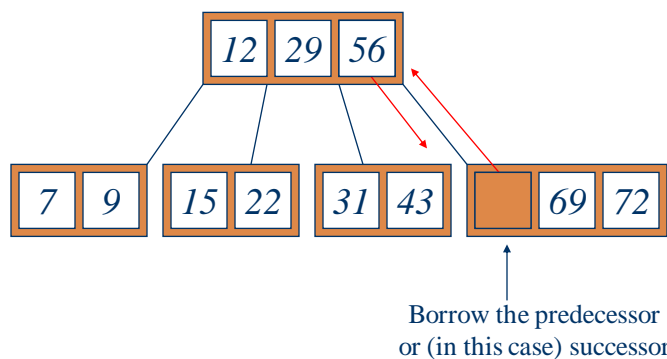


Xóa 2: Nút lá vẫn còn đủ số lượng phần tử theo yêu cầu

[www.themegallery.com](http://www.themegallery.com)



## Trường hợp: Xóa khóa không ở nút lá



www.themegallery.com



## B-tree: Cân bằng lại cây sau khi xóa

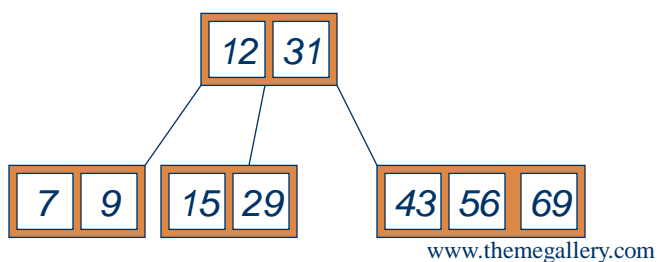
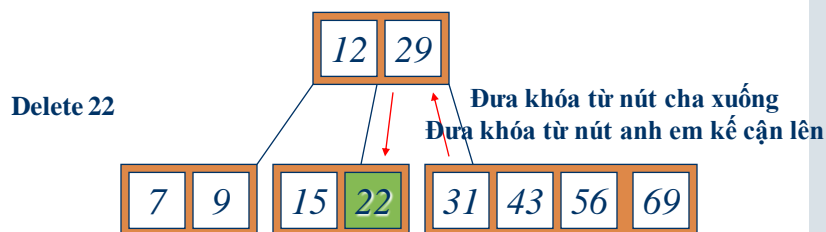
- ❖ Nếu một trong các nút anh em kế cận nút đang xét có số lượng khóa nhiều hơn số lượng tối thiểu
  - Đưa một khóa của nút anh em lên nút cha.
  - Đưa một khóa ở nút cha xuống nút đang xét.
- ❖ Nếu tất cả các nút anh em kế cận nút đang xét đều có số lượng khóa vừa đủ số lượng tối thiểu.
  - Chọn một 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.

www.themegallery.com



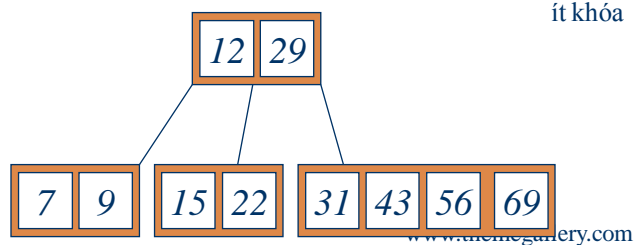
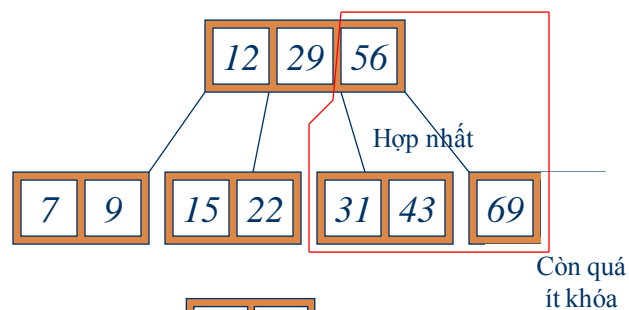
## Trường hợp:

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





## Bài tập luyện tập

❖ Cho B-tree bậc 5 gồm các khóa sau (chèn vào theo thứ tự):  
**3, 7, 9, 23, 45, 1, 5, 14, 25, 24, 13, 11, 8, 19, 4, 31, 35, 56**

❖ Thêm khóa: **2, 6, 12, 0, 10, 11**

❖ Xóa khóa: **4, 5, 7, 3, 14**

[www.themegallery.com](http://www.themegallery.com)

**ĐHSP**



# Thank You!