CẤU TRÚC DỮ LIỆU
CÂY M-NHÁNH VS B CÂY

Bùi Tiến Lên

01/01/2017

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

CÂY M-NHÁNH

Cây m-nhánh

Định nghĩa 1

Cây *m*-nhánh (*m-way tree*) là một cây tìm kiếm có những tính chất sau

- Mỗi nút có
  - tối thiểu 1 khóa
  - tối đa $m-1$ khóa có giá trị phân biệt
- Các khóa trong mỗi nút được sắp thứ tự tăng dần

Cây m-nhánh (cont.)

Định nghĩa 1

- Mỗi nút có k khóa $\{v_1, ..., v_k\}$ thì sẽ có $k+1$ cây con $\{T_1, ..., T_{k+1}\}$, các cây con có thể **rỗng**
  - Cây con **đầu** T_1 sẽ chứa các khóa v trong khoảng

$$v \in (-\infty, v_1) \tag{1}$$

  - Cây con **cuối** T_{k+1} sẽ chứa các khóa v trong khoảng

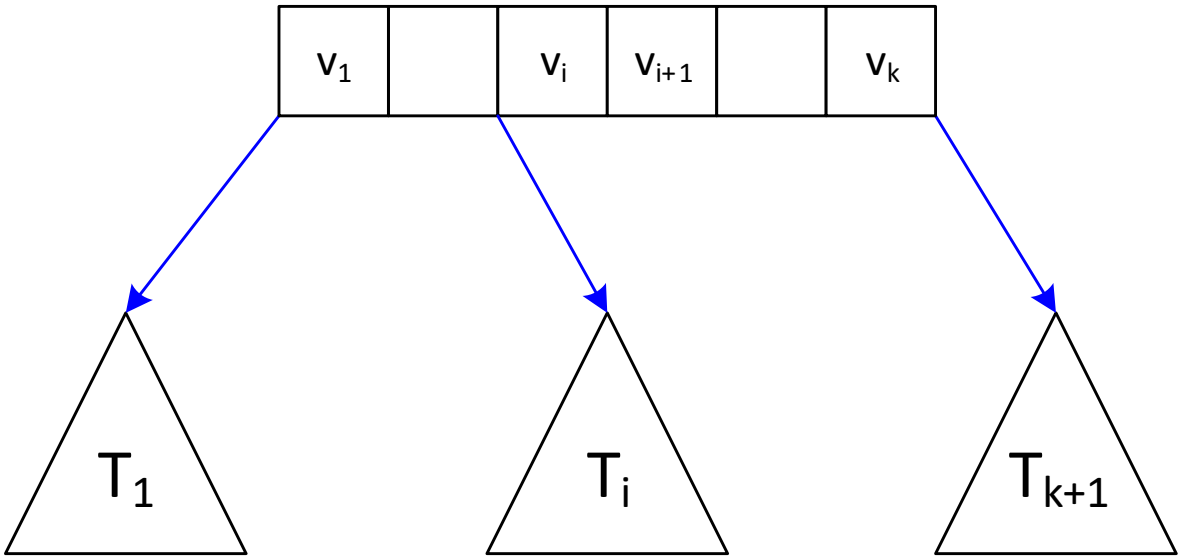
$$v \in (v_k, \infty) \tag{2}$$

  - Cây con $T_i, i = 2, .., k$ sẽ chứa các khóa v trong khoảng

$$v \in (v_i, v_{i+1}) \tag{3}$$

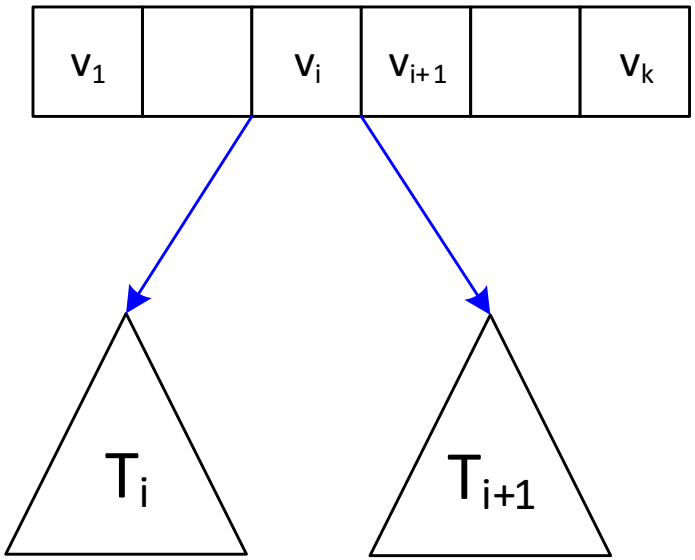
  - Mỗi khóa v_i sẽ có cây con trái là T_i và cây con phải T_{i+1}

Minh họa



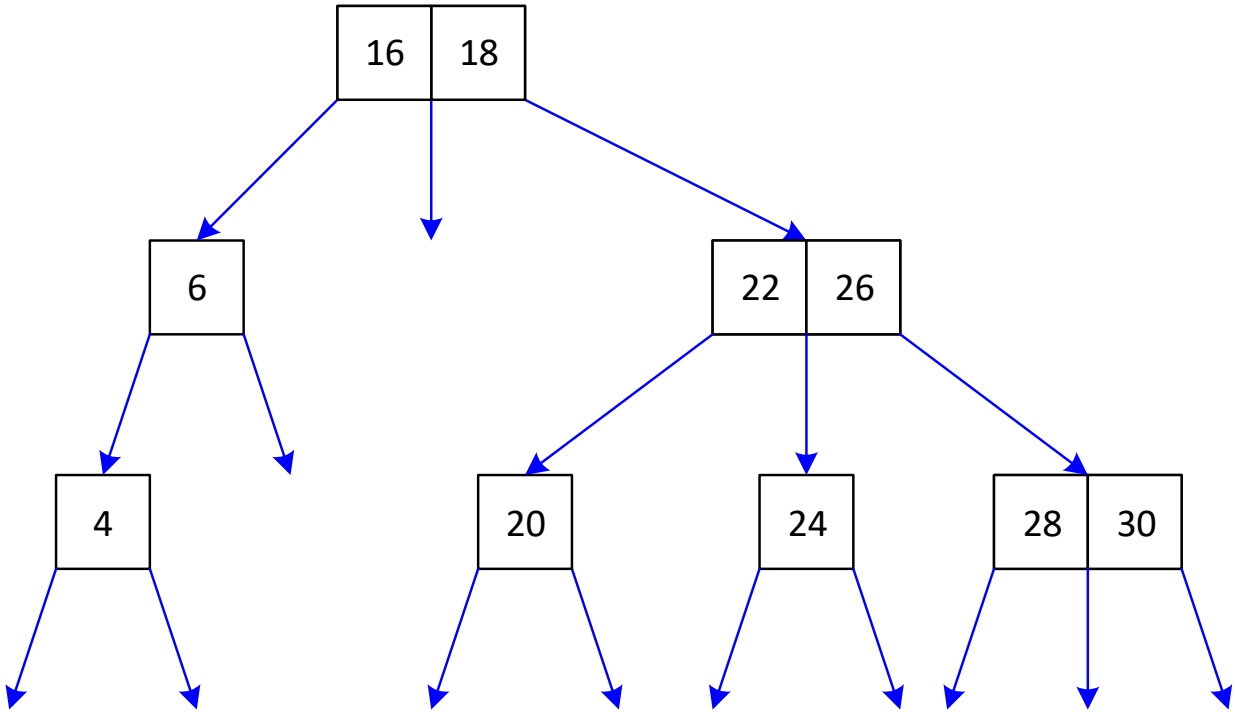
Hình 1: Nút và các khóa và các cây con

Minh họa (cont.)



Hình 2: Khóa và cây con trái và con phải

Minh họa (cont.)



Hình 3: Cây 3-nhánh

Các thao tác trên cây m-nhánh

Đối với cây m-nhánh có các thao tác cơ bản trên cây

- **Duyệt** từng khóa của cây
- **Tìm** một khóa trong cây
- **Thêm** một khóa vào cây
- **Xóa** một khóa khỏi cây

Thao tác duyệt cây

Ta có thể xem cây như một đồ thị tổng quát và áp dụng các thuật toán duyệt của đồ thị để duyệt cây. Có hai thuật toán duyệt cơ bản

- Duyệt theo chiều sâu (*Depth First Traversal - DFT*)
- Duyệt theo chiều rộng (*Breath First Traversal - BFT*)

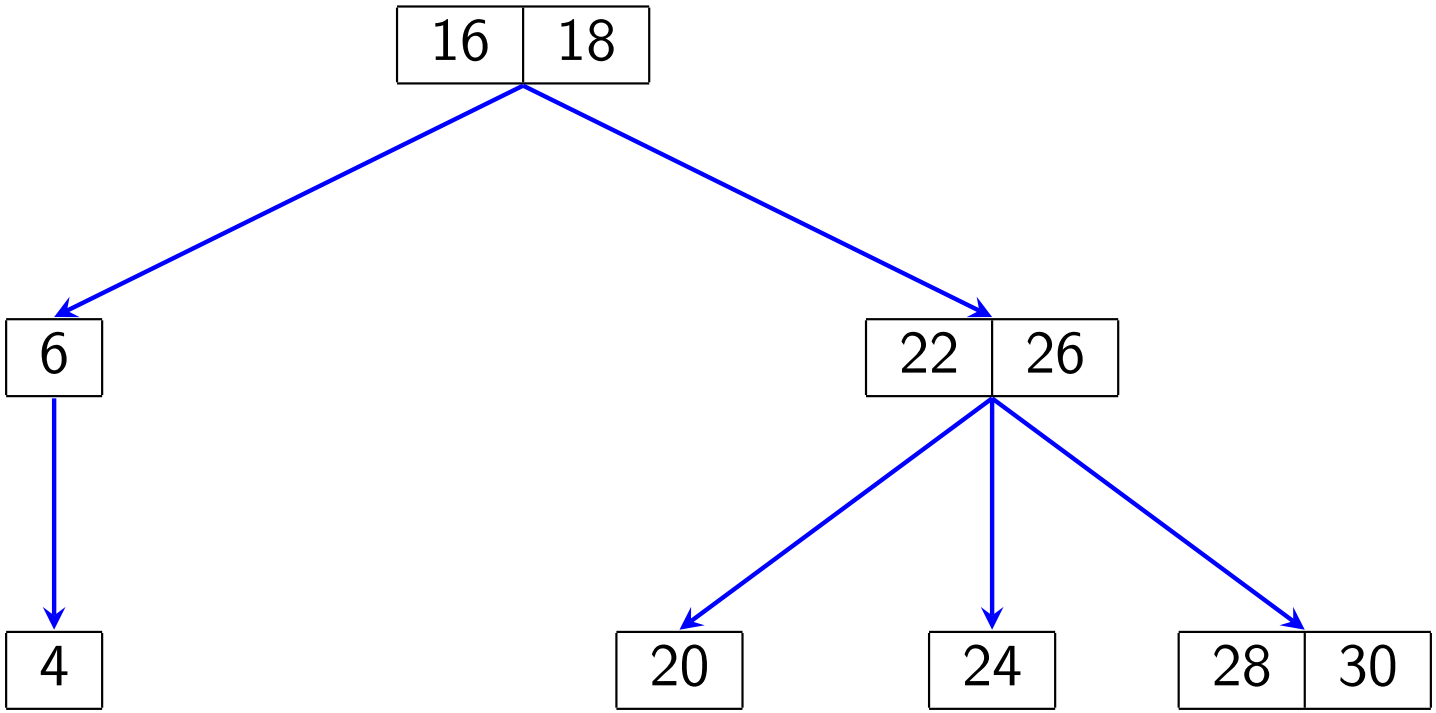
Duyệt theo chiều sâu

```
PROCEDURE Dft(r)
BEGIN
  Thăm nút r
  FOR mỗi nút con u của r DO
    IF u chưa được thăm THEN DFT(u)
END
```

Duyệt theo chiều rộng

```
PROCEDURE BFT(r)
BEGIN
  Đưa nút r vào hàng đợi queue
  WHILE queue khác rỗng
  BEGIN
    Lấy nút đỉnh khỏi queue gọi là nút x
    Thăm nút x
    FOR mỗi nút con u của x DO
      IF u chưa thăm THEN đưa u vào queue
  END
END
```

Minh họa duyệt cây m-nhánh



Hình 4: Hãy xác định khóa lớn nhất và nhỏ nhất của cây. Hãy xác định khóa đứng trước và đứng sau của khóa 18

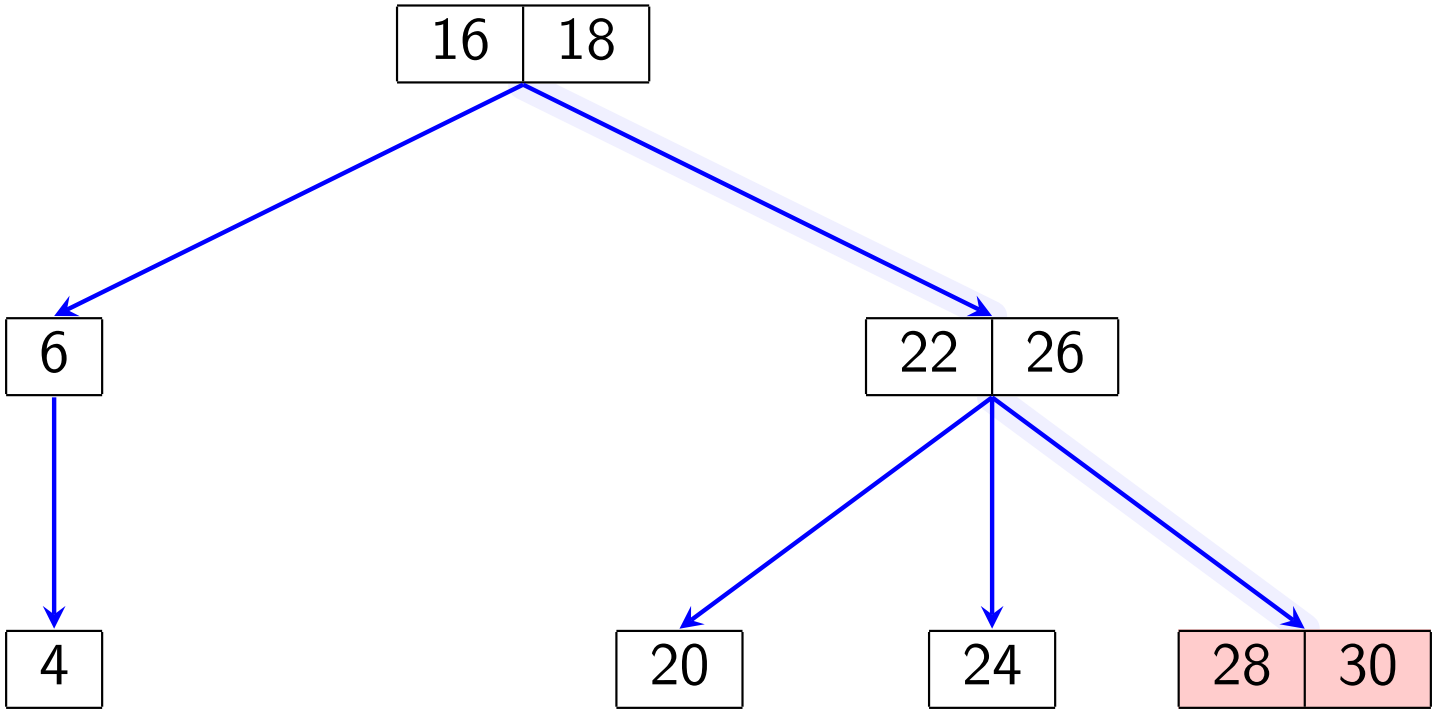
Tìm kiếm một khóa trong cây

Tìm kiếm trên cây m-nhánh tương tự như tìm kiếm trên phân cây nhị phân tìm kiếm

- Bắt đầu từ nút gốc của cây
- Duyệt cây theo hướng từ trên xuống (*top-down*)
- Tại mỗi nút so sánh khóa cần tìm với các giá trị khóa tại nút (có thể sử dụng phương pháp tìm kiếm nhị phân)
- Nếu tìm thấy thì dừng việc tìm kiếm lại nếu không thì sử dụng các phương trình (1, 2, 3) để xác định cây con có khả năng chứa khóa và tiếp tục tìm trong cây con của nút này

Minh họa tìm khóa trên cây

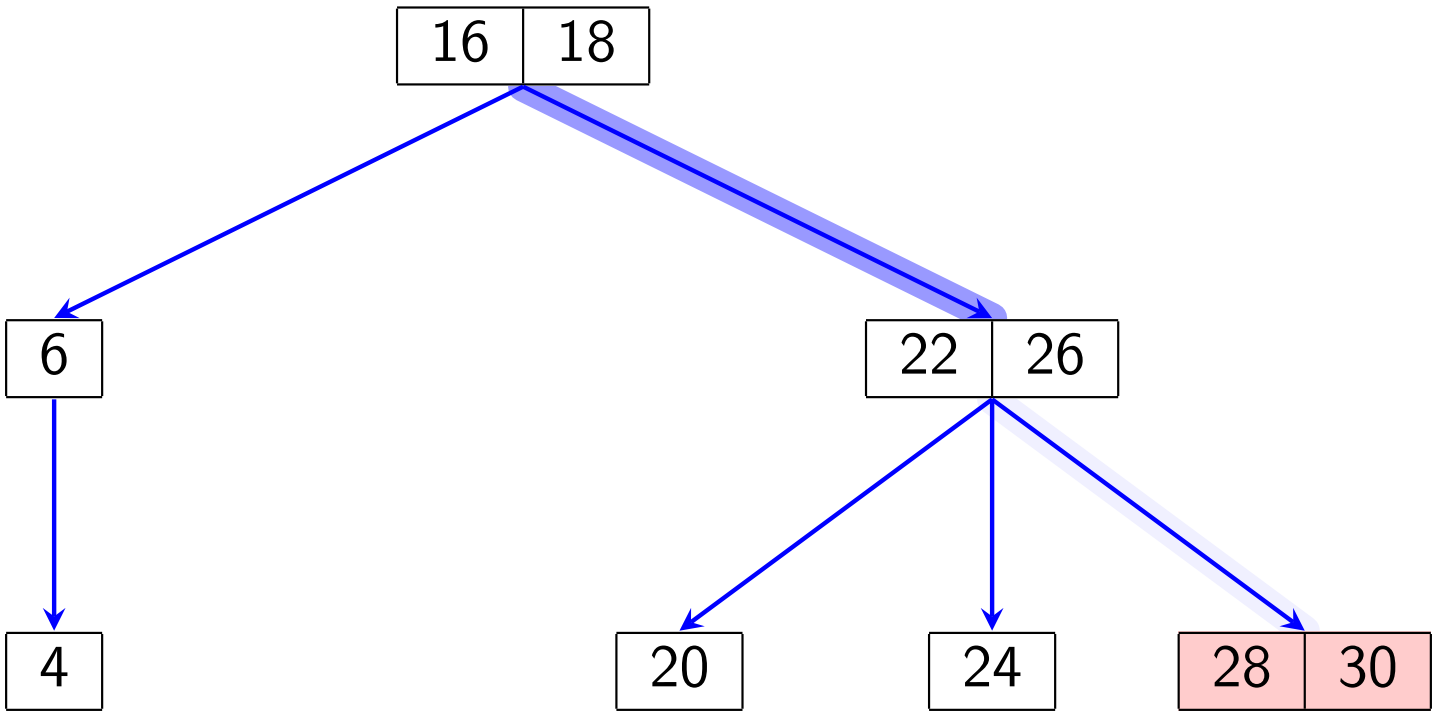
- Tìm kiếm khóa 30



Hình 5: Tìm kiếm

Minh họa tìm khóa trên cây

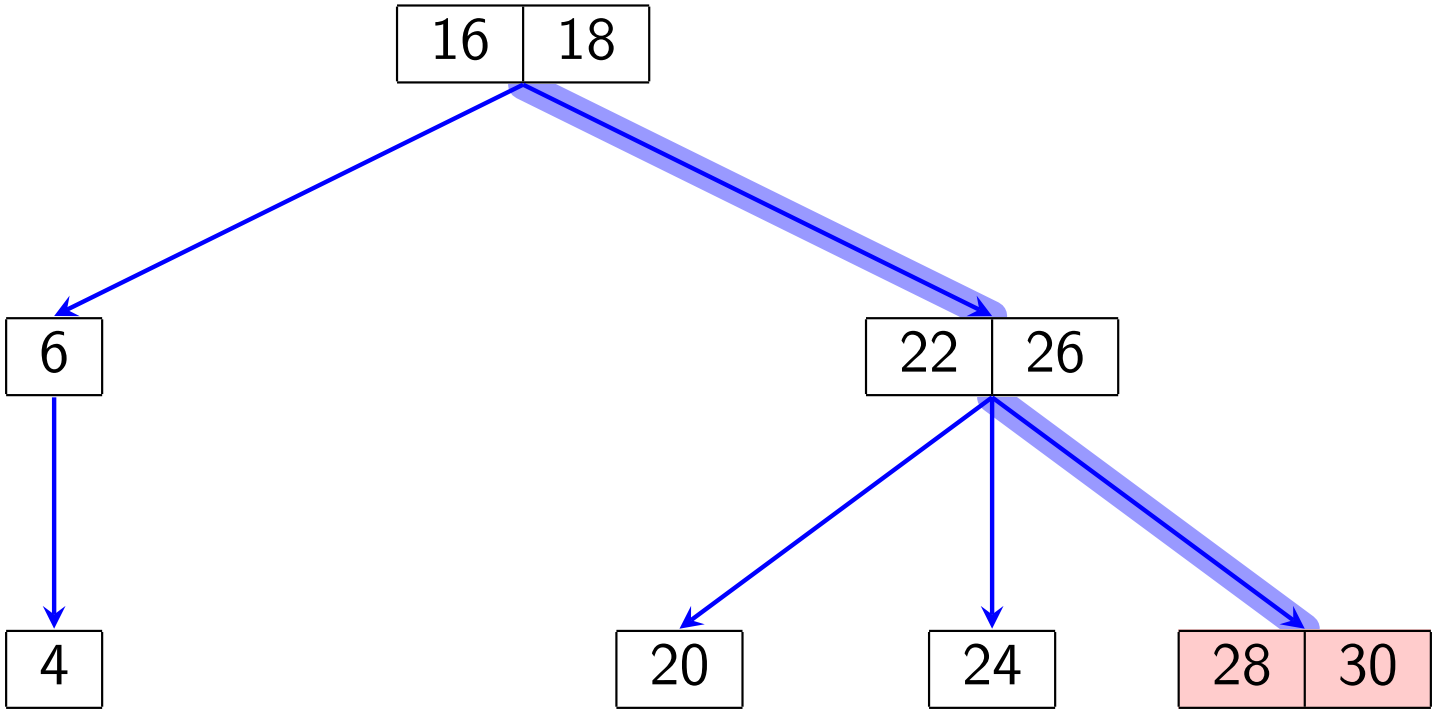
- Tìm kiếm khóa 30



Hình 5: Tìm kiếm

Minh họa tìm khóa trên cây

- Tìm kiếm khóa 30



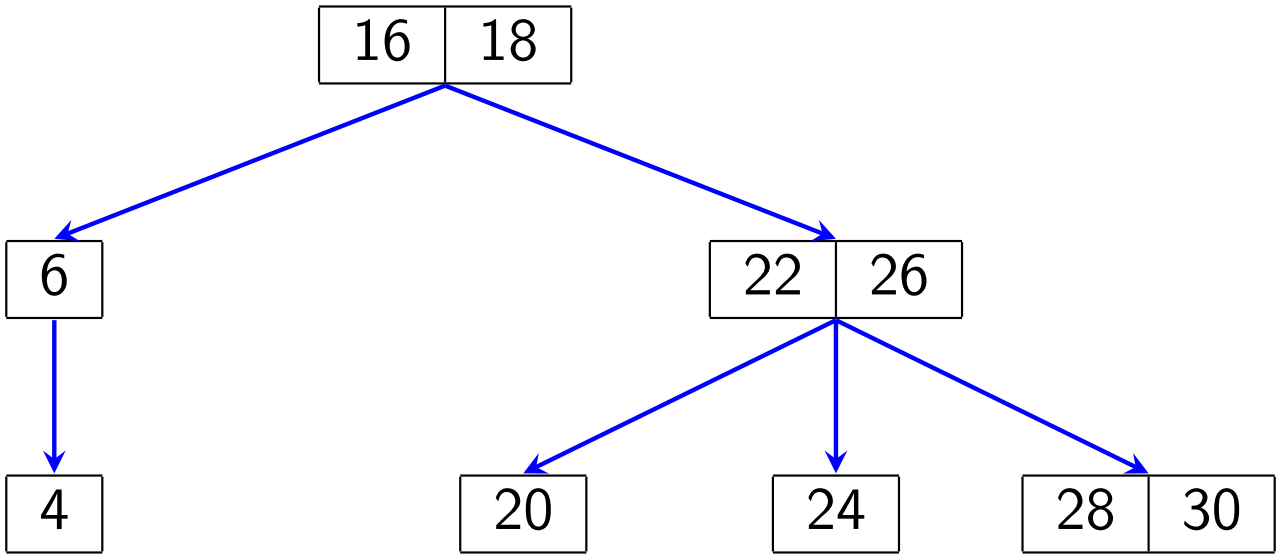
Hình 5: Tìm kiếm

Thao tác thêm khóa vào cây

Thêm một khóa v vào cây m-nhánh

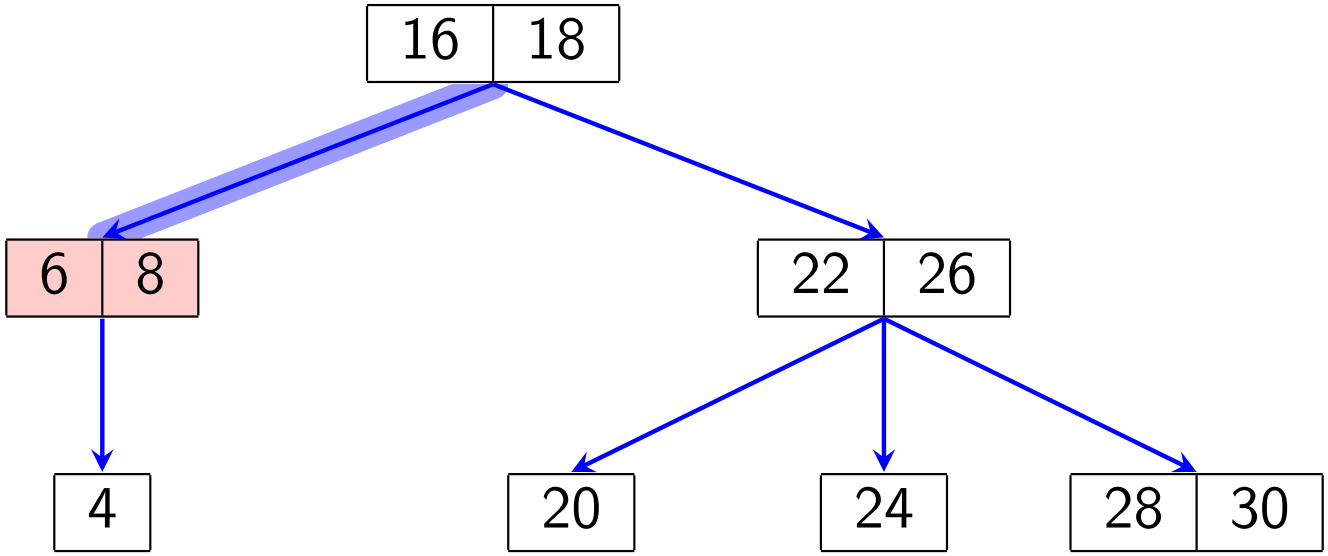
- Duyệt cây để tìm kiếm vị trí của v cho đến khi gặp cây con rỗng
- Thêm khóa v vào nút cha của cây con rỗng nếu nút cha còn "chỗ trống" (ít hơn $m - 1$ khóa)
- Hoặc, nếu không còn nút trống tạo nút mới và thêm khóa v vào nút đó

Minh họa thêm khóa vào cây



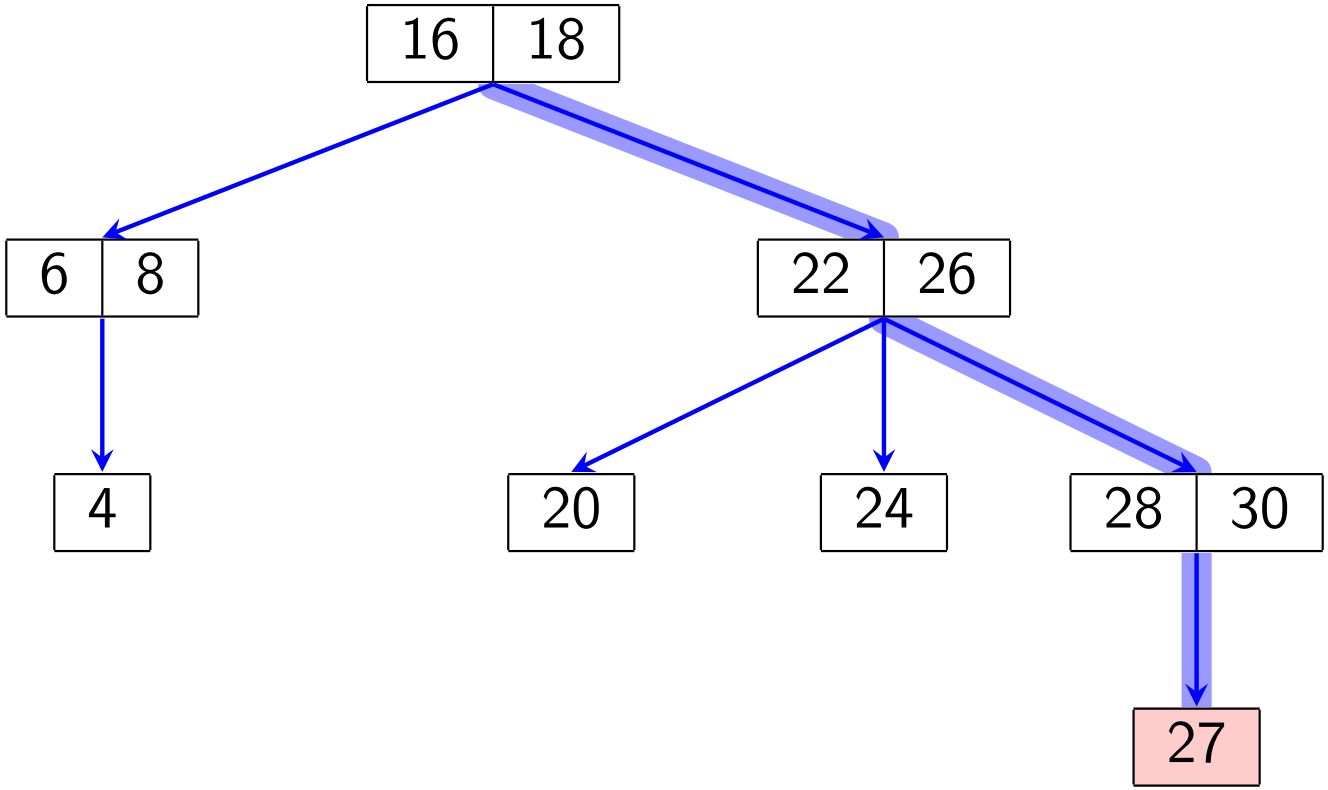
Hình 6: Cây 3-nhánh

Minh họa thêm khóa vào cây (cont.)



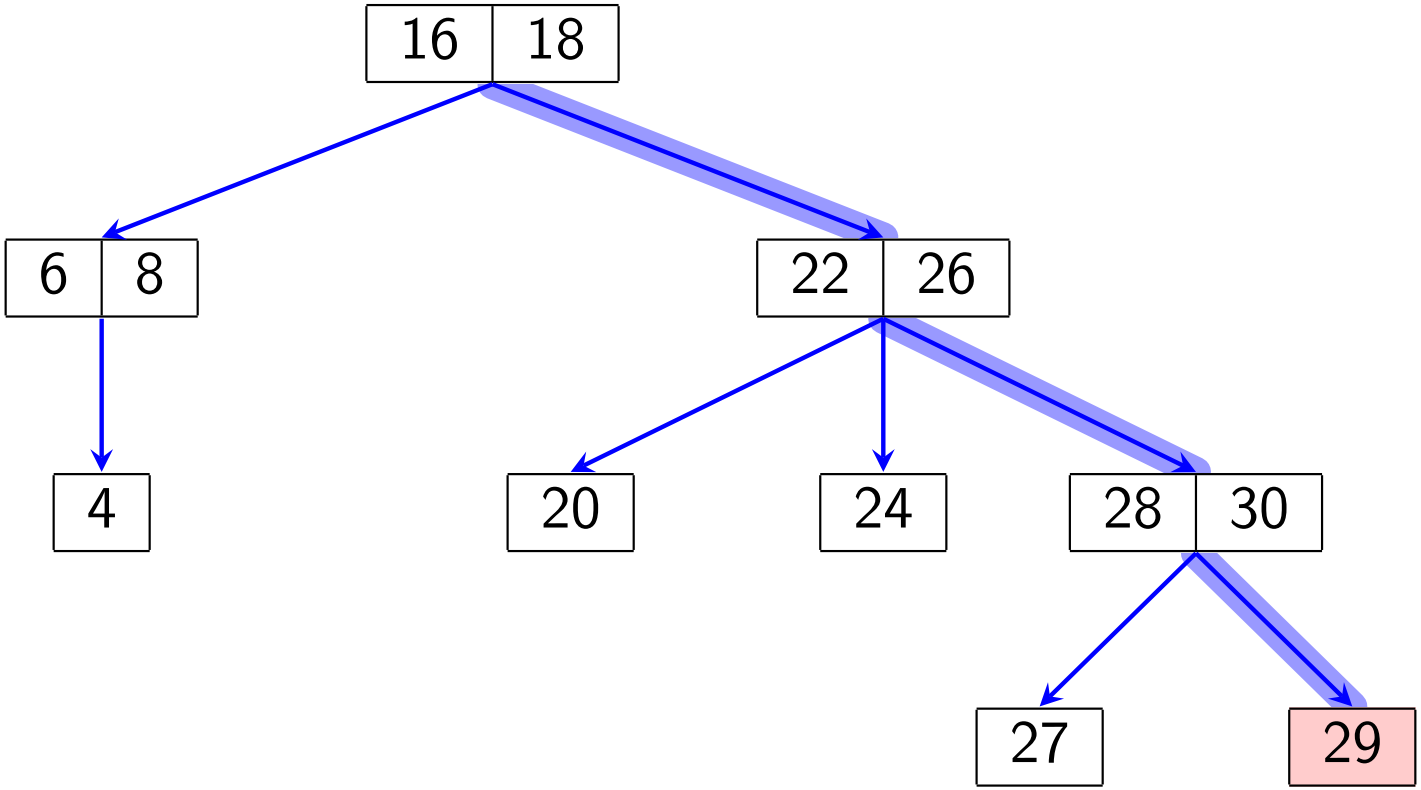
Hình 7: Thêm khóa 8

Minh họa thêm khóa vào cây (cont.)



Hình 8: Thêm khóa 27

Minh họa thêm khóa vào cây (cont.)



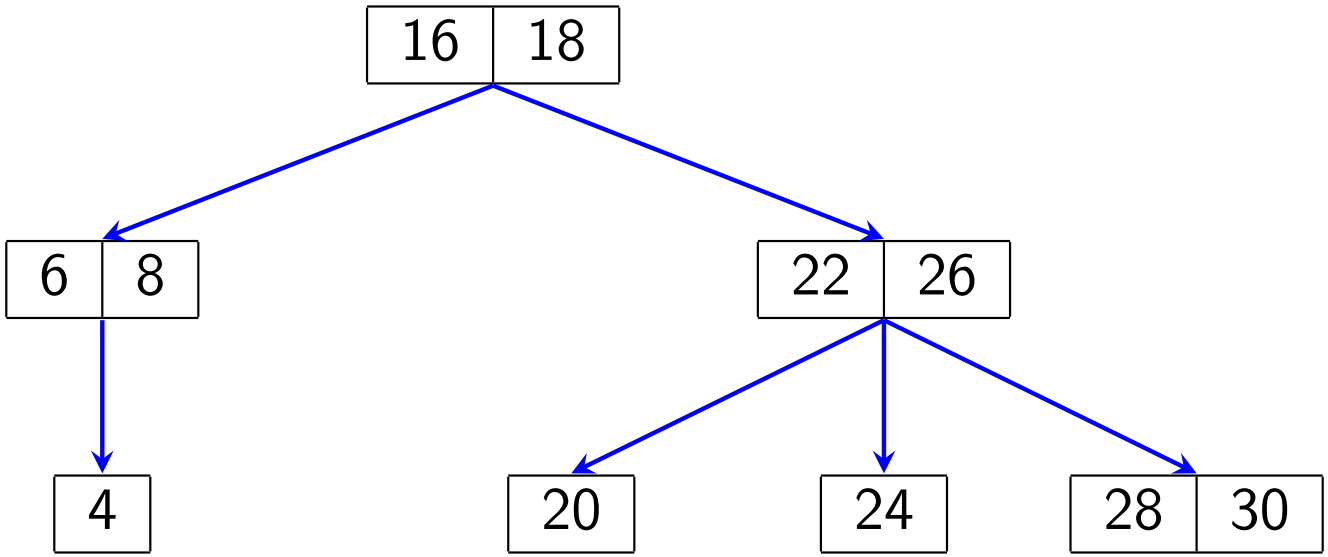
Hình 9: Thêm khóa 29

Xóa khóa khỏi cây

Xóa một khóa hay phần tử v ra khỏi cây

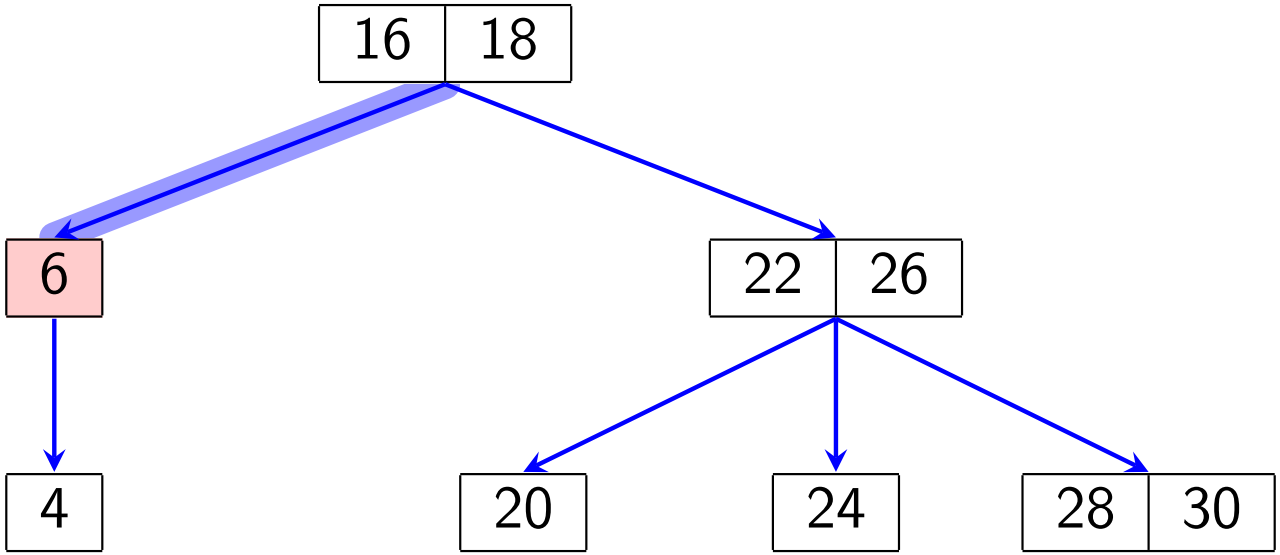
- Nếu v nằm giữa 2 cây con rỗng thì xóa v
- Nếu v có cây con thay thế v bằng:
  - Phần tử **lớn nhất** của cây con bên trái
  - Hoặc phần tử **bé nhất** của cây con bên phải

Minh họa xóa khóa khỏi cây



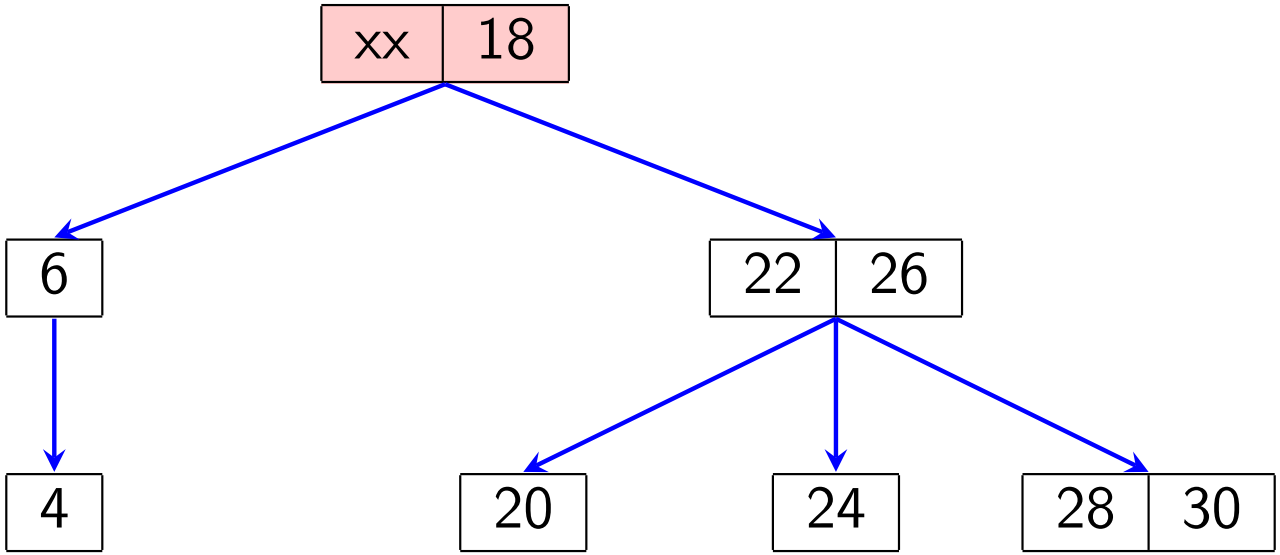
Hình 10: Cây 3-nhánh

Minh họa xóa khóa khỏi cây (cont.)



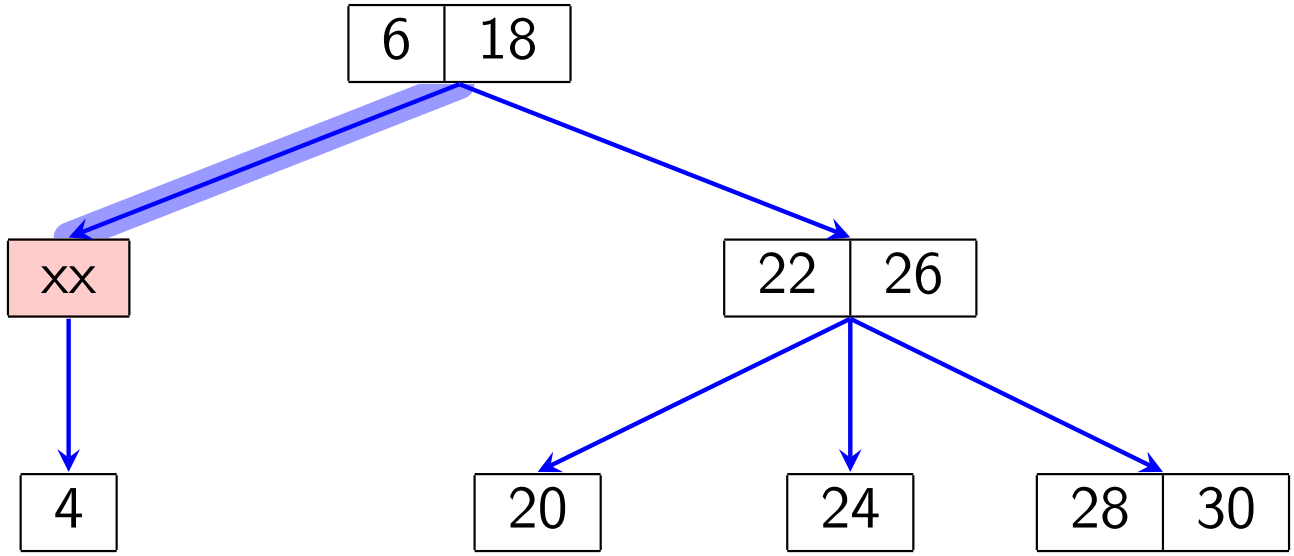
Hình 11: Xóa khóa 8

Minh họa xóa khóa khỏi cây (cont.)



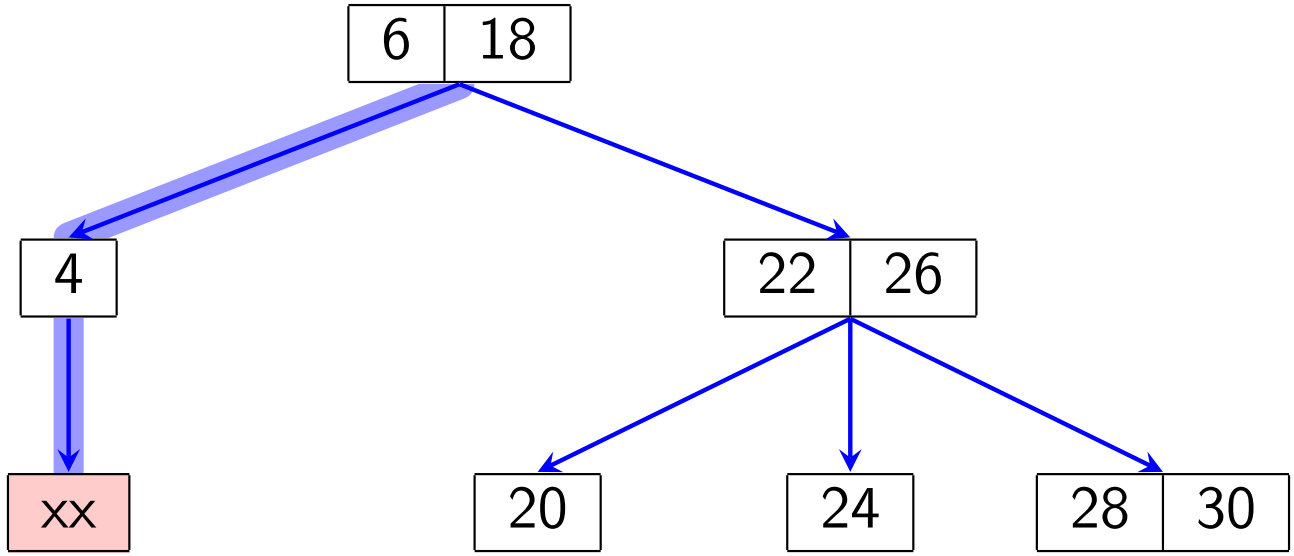
Hình 12: Xóa khóa 16: xóa khóa 16

Minh họa xóa khóa khỏi cây (cont.)



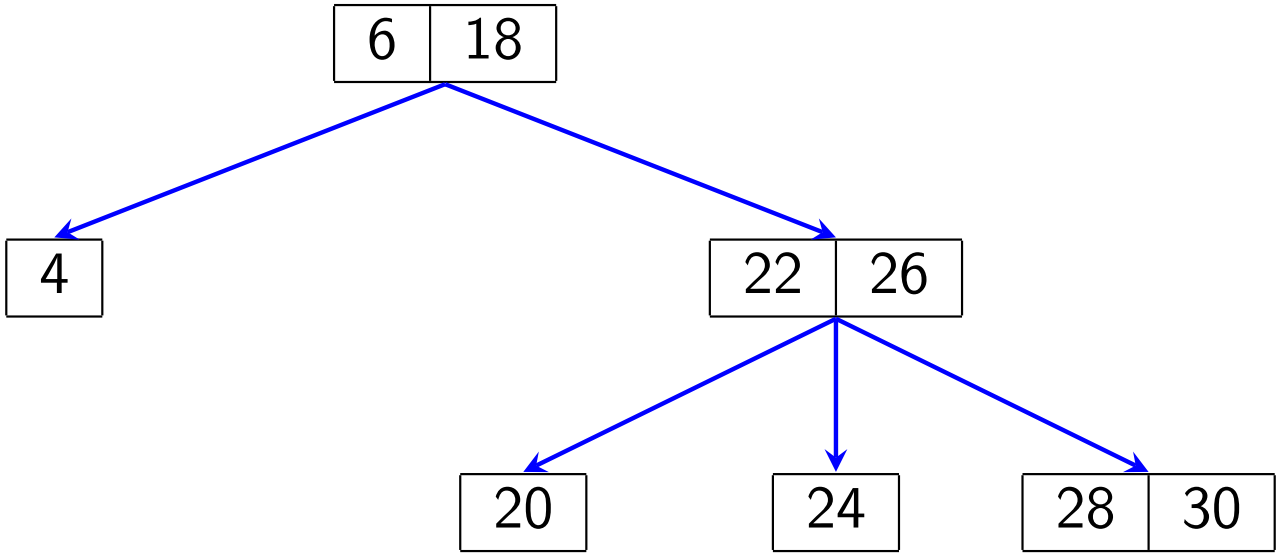
Hình 13: Xóa khóa 16: khóa 6 thay thế khóa 16, xóa khóa 6

Minh họa xóa khóa khỏi cây (cont.)



Hình 14: Xóa khóa 16: khóa 4 thay thế khóa 6, xóa khóa 4

Minh họa xóa khóa khỏi cây (cont.)



Hình 15: Xóa khóa 16: kết quả

Bài luyện tập

Ví dụ 1

Hãy xây dựng cây 3-nhánh từ dãy {4, 3, 5, 1, 2, 7, 9, 8, 15, 11, 12,16}

- Xóa các nút 8, 16
- Thêm các nút 6, 17

B-CÂY

B-cây

Giới thiệu

- B-cây được Rudolf Bayer, Edward M. McCreight phát minh là cây cân bằng tổng quát.
- B-cây phù hợp cho các hệ thống đọc và ghi dữ liệu lớn. Nó thường được dùng trong các cơ sở dữ liệu và hệ thống tập tin. B-cây sử dụng tối thiểu các thao tác truy xuất đĩa
- Có thể quản lý số phần tử rất lớn

B-cây (cont.)

Định nghĩa 2

B-cây (*B-tree*) là một cây m-nhánh ($m > 2$) và thỏa

- Nút gốc có ít nhất 1 khóa
- Các nút không phải gốc có ít nhất $\left\lfloor \frac{m-1}{2} \right\rfloor$ khóa
- Tất cả các nút lá đều có cùng một mức (điều kiện cân bằng)

B-cây (cont.)

Ký hiệu và một số B-cây thông dụng

B-cây m-nhánh được ký hiệu

- B-cây bậc m
- Hoặc cây $\left(\left\lfloor \frac{m-1}{2} \right\rfloor + 1, m\right)$

Một số B-cây phổ biến

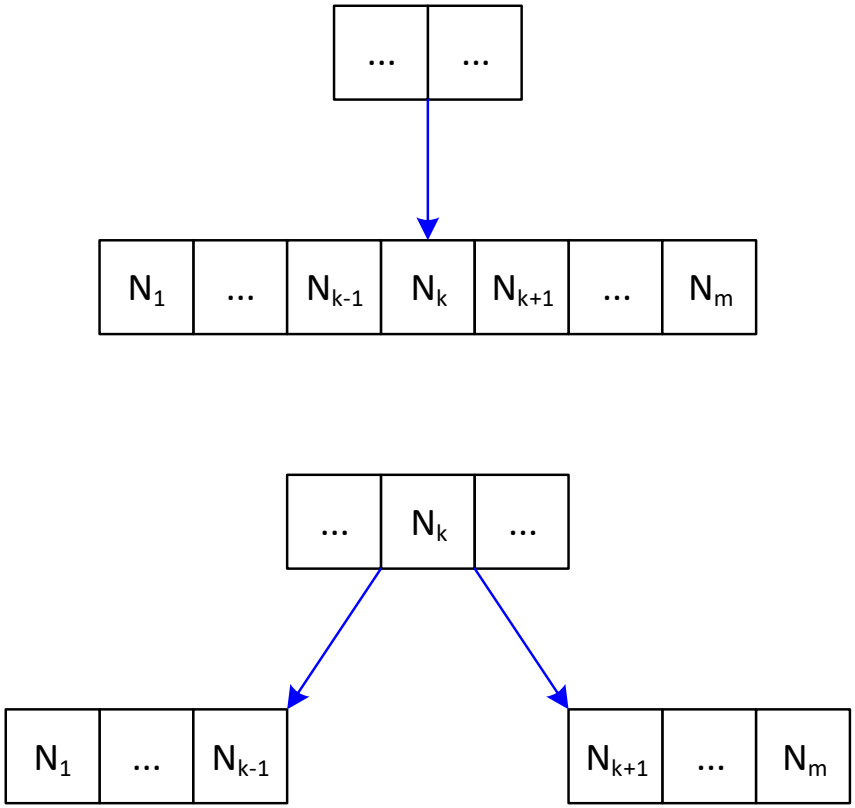
- B-cây bậc 3 hoặc cây (2, 3) hoặc cây 2-3
- B-cây bậc 4 hoặc cây (2, 4) hoặc cây 2-3-4

Thao tác thêm một phần tử

Thêm một khóa v vào B-cây có m-nhánh

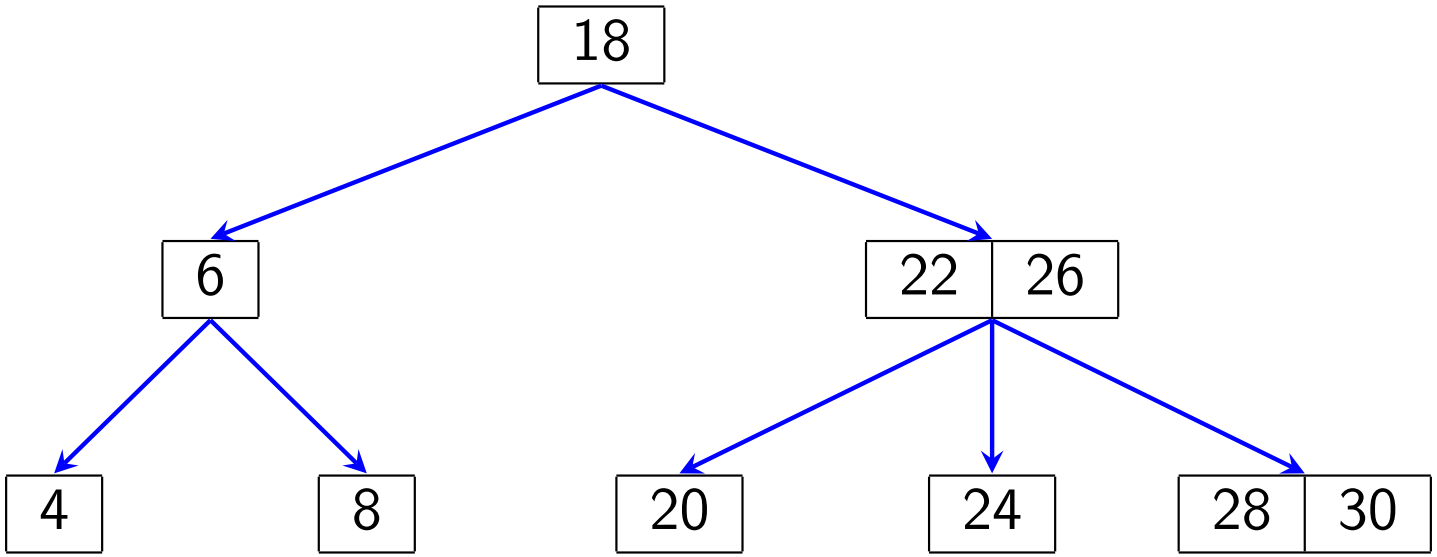
- Thêm khóa v vào một nút lá N phù hợp
- **Nếu** nút lá vừa thêm vào bị "tràn" khóa (có nhiều hơn $m-1$ khóa) **thì**
  - "Tách nút" N thành hai nút
  - Và chuyển khóa "ở giữa" lên nút cha

Thao tác thêm một phần tử (cont.)



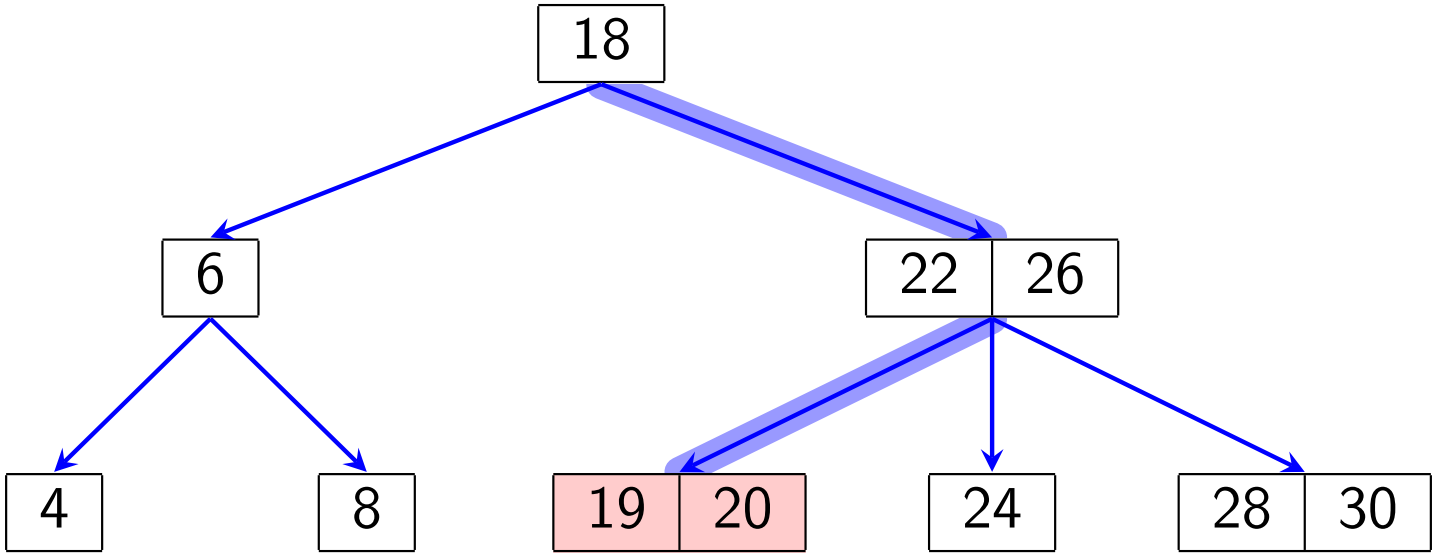
Hình 16: Xử lý "tách nút"

Minh họa thao tác thêm phần tử



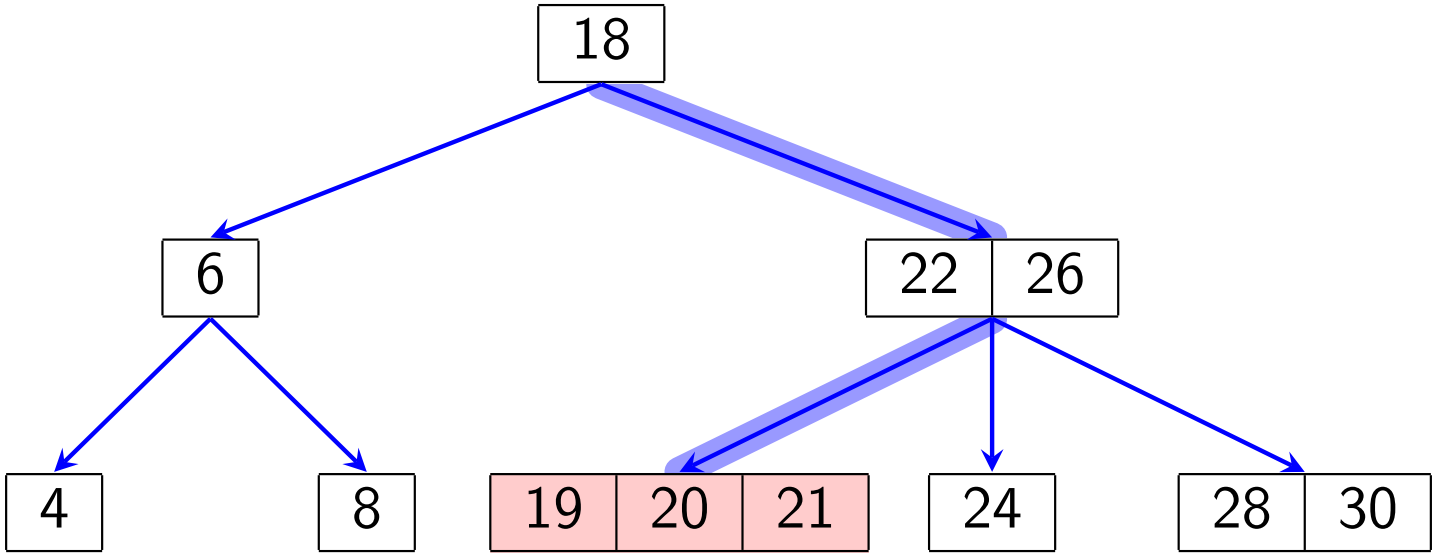
Hình 17: B-cây 3-nhánh

Minh họa thao tác thêm phần tử (cont.)



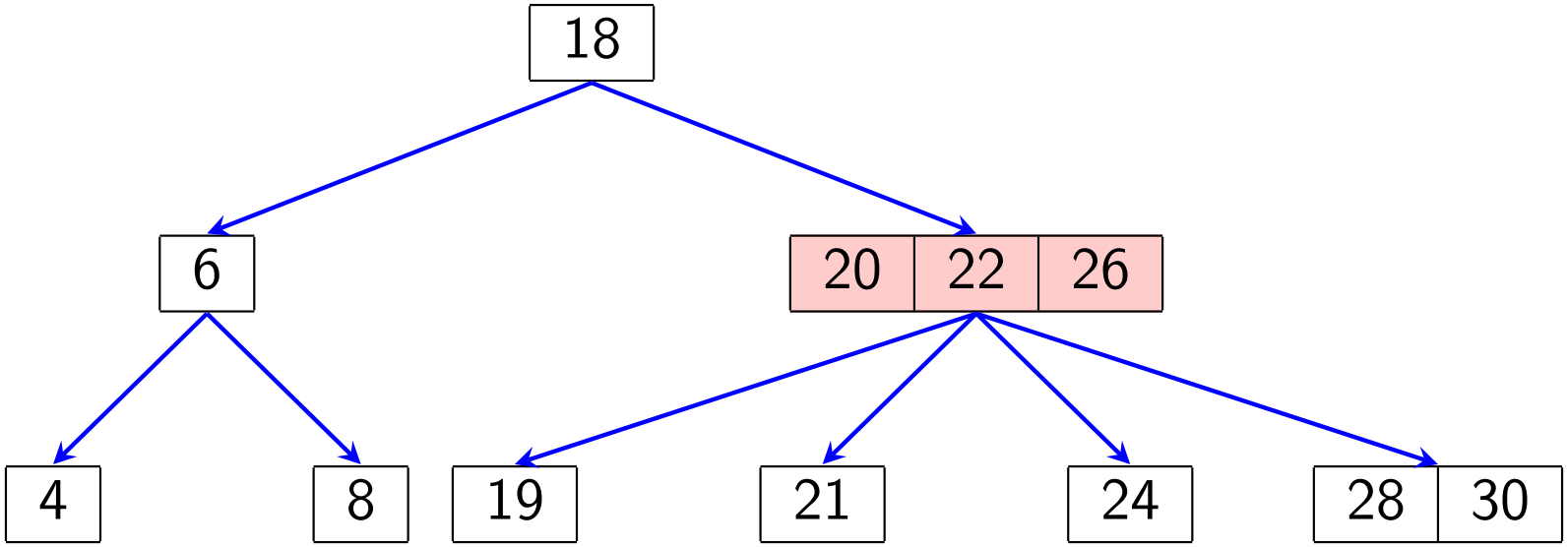
Hình 18: Thêm khóa 19

Minh họa thao tác thêm phần tử (cont.)



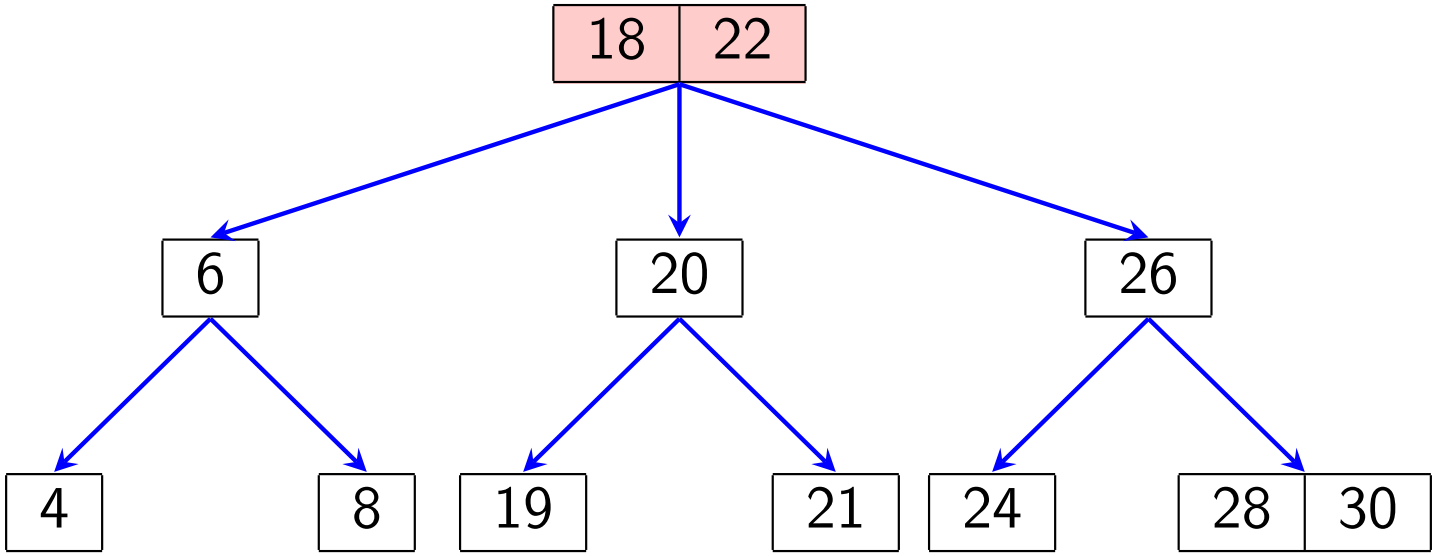
Hình 19: Thêm khóa 21: thêm khóa 21 và nút bị tràn

Minh họa thao tác thêm phần tử (cont.)



Hình 20: Thêm khóa 21: tách thành hai nút và đưa khóa 20 lên nút cha và nút cha bị tràn

Minh họa thao tác thêm phần tử (cont.)



Hình 21: Thêm khóa 21: tách nút cha thành hai nút và đưa khóa 22 lên nút ông

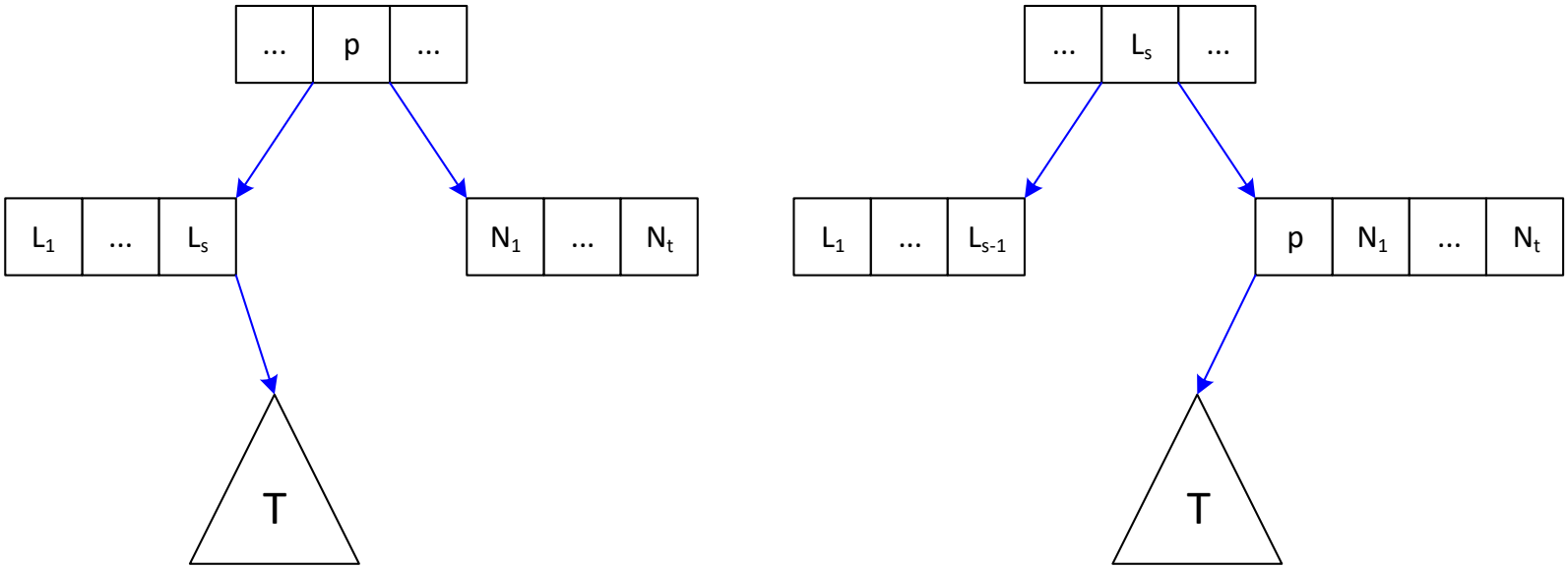
Xóa một khóa khỏi cây

- Xóa khóa v khỏi nút N
- **Nếu** nút N sau khi xóa bị "hụt" khóa (có ít hơn $\left\lfloor \frac{m-1}{2} \right\rfloor$ khóa) thì
  - Trường hợp 1: Nếu nút anh em kế bên có dư khóa (nhiều hơn $\left\lfloor \frac{m-1}{2} \right\rfloor$ khóa) thì "mượn khóa"
  - Trường hợp 2: Nếu nút anh em kế bên không dư khóa thì "nhập nút" với một nút anh em

Xóa một khóa khỏi cây (cont.)

- **Trường hợp 1**:
  - Nút $\boldsymbol{N} = \{N_1, ..., N_t\}$ thiếu khóa
  - Nút $\boldsymbol{L} = \{L_1, ..., L_s\}$ "anh em bên trái" dư khóa
  - $\boldsymbol{N}$ là nút gốc của cây con phải của khóa p và $\boldsymbol{L}$ là nút gốc cây con trái của p

Xóa một khóa khỏi cây (cont.)

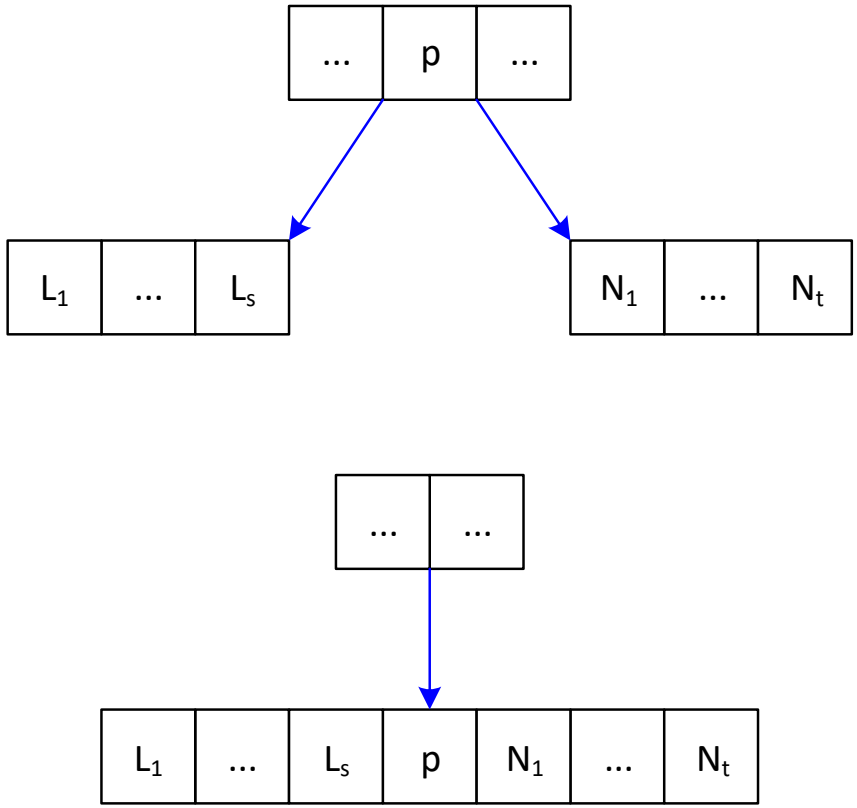


Hình 22: Xử lý "mượn khóa"

Xóa một khóa khỏi cây (cont.)

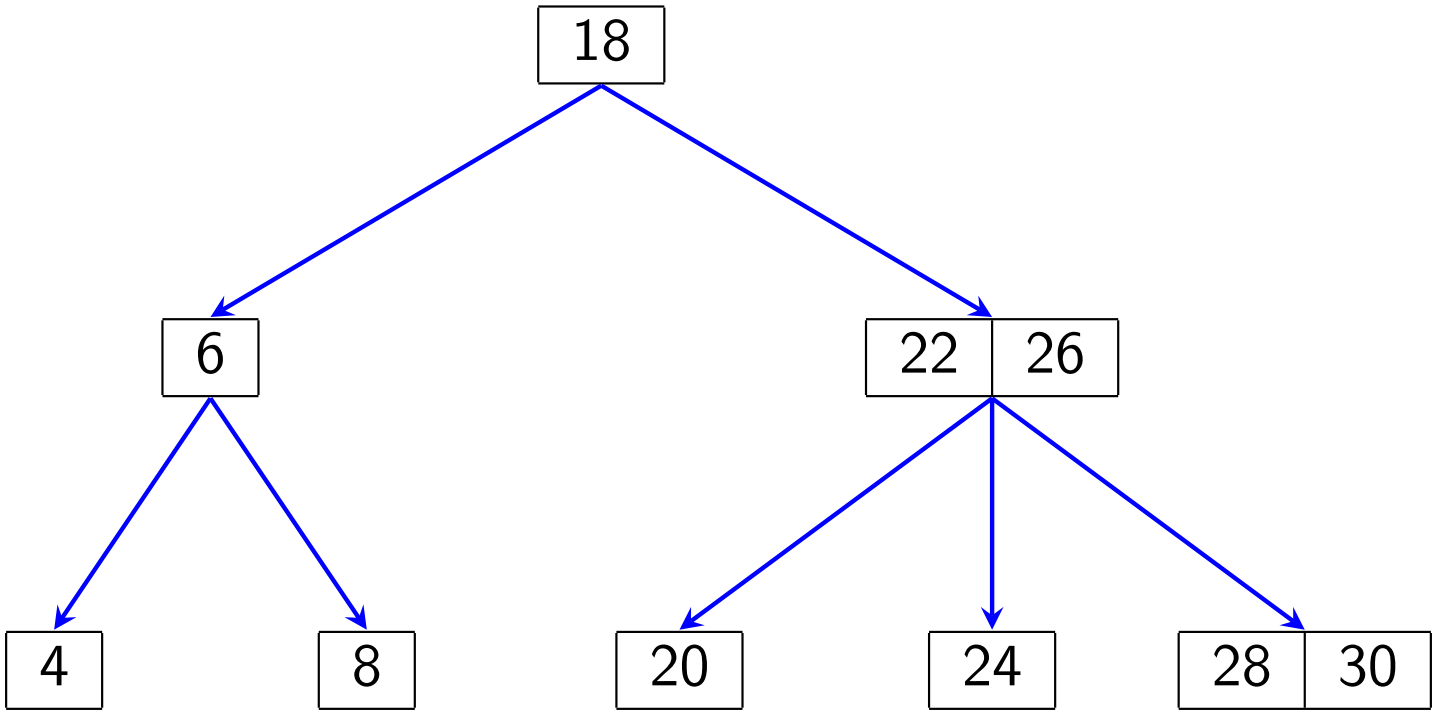
- **Trường hợp 2**:
  - Nút $\boldsymbol{N} = \{N_1, ..., N_t\}$ thiếu khóa và không có nút anh em dư khóa
  - Nút $\boldsymbol{L} = \{L_1, ..., L_s\}$
  - $\boldsymbol{N}$ là nút gốc của cây con phải của khóa p và $\boldsymbol{L}$ là nút gốc cây con trái của p

Xóa một khóa khỏi cây (cont.)



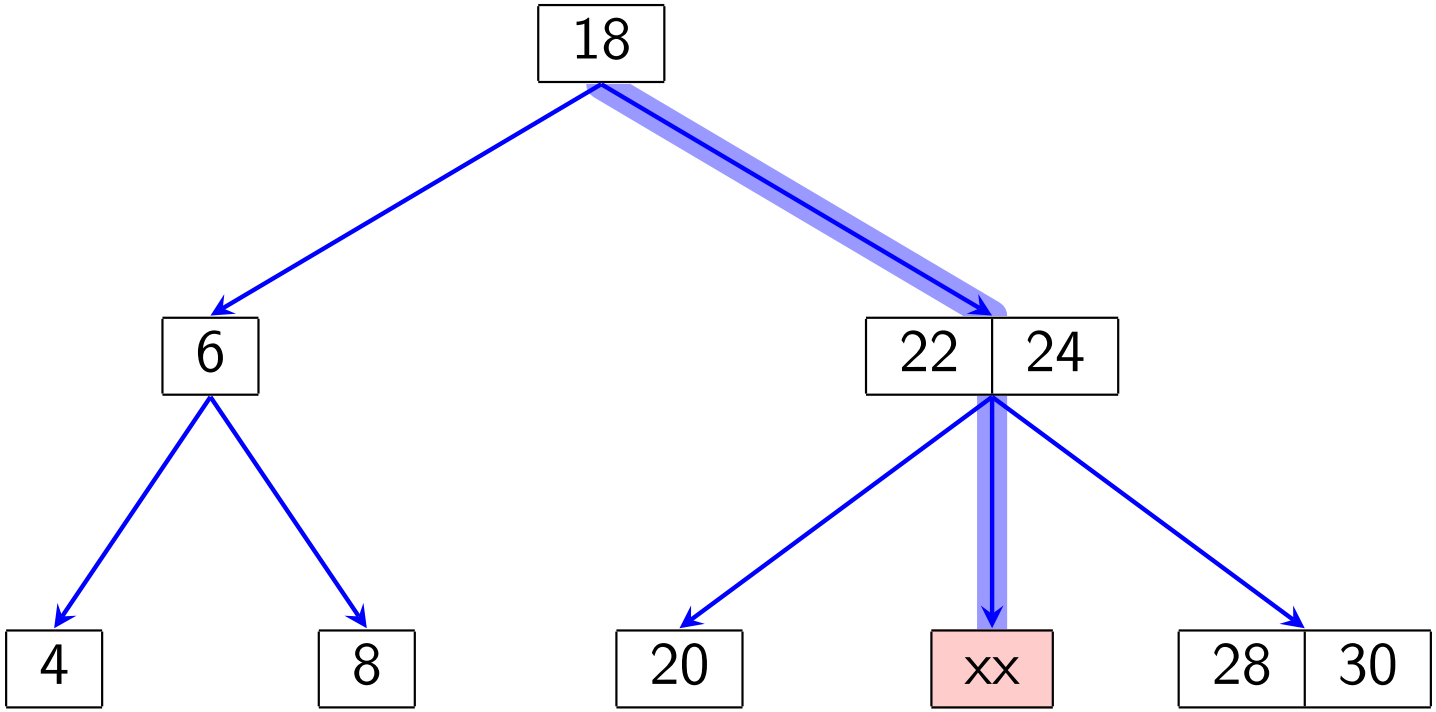
Hình 23: Xử lý "nhập nút"

Minh họa xóa phần tử



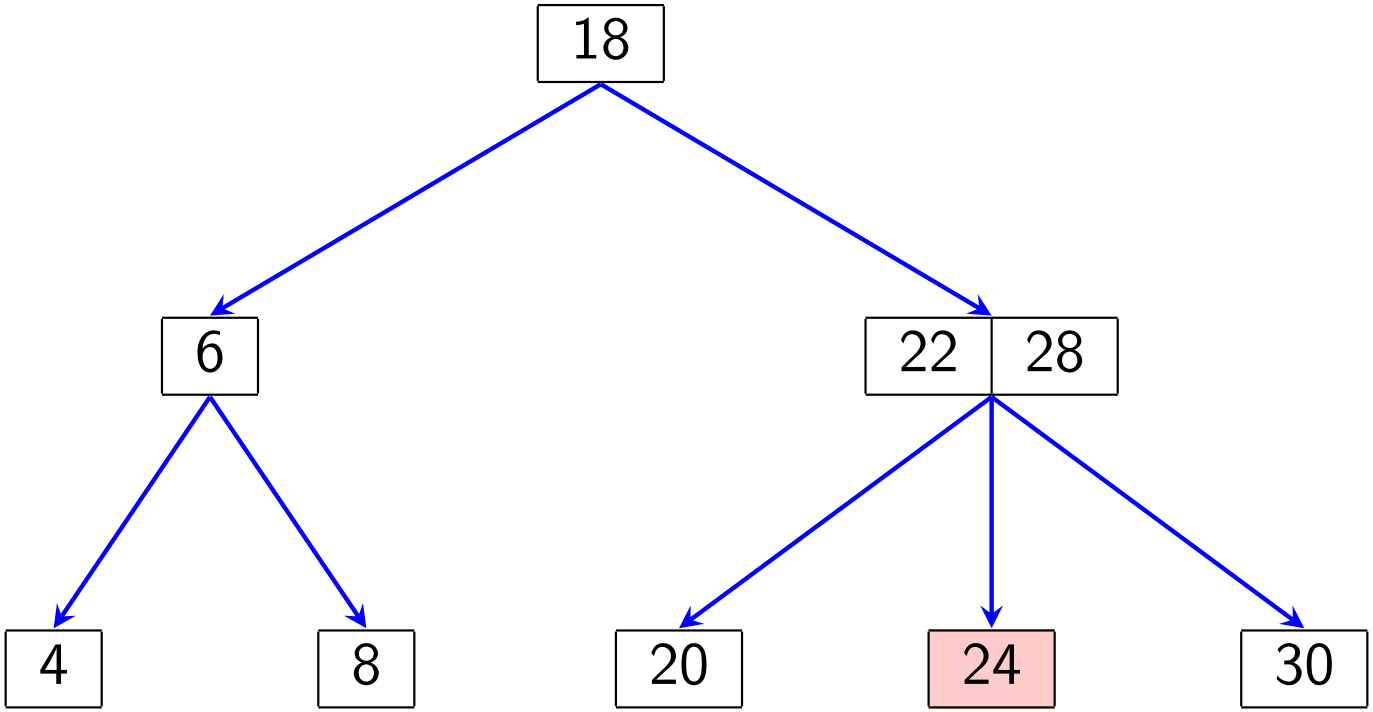
Hình 24: B-cây 3-nhánh

Minh họa xóa phần tử (cont.)



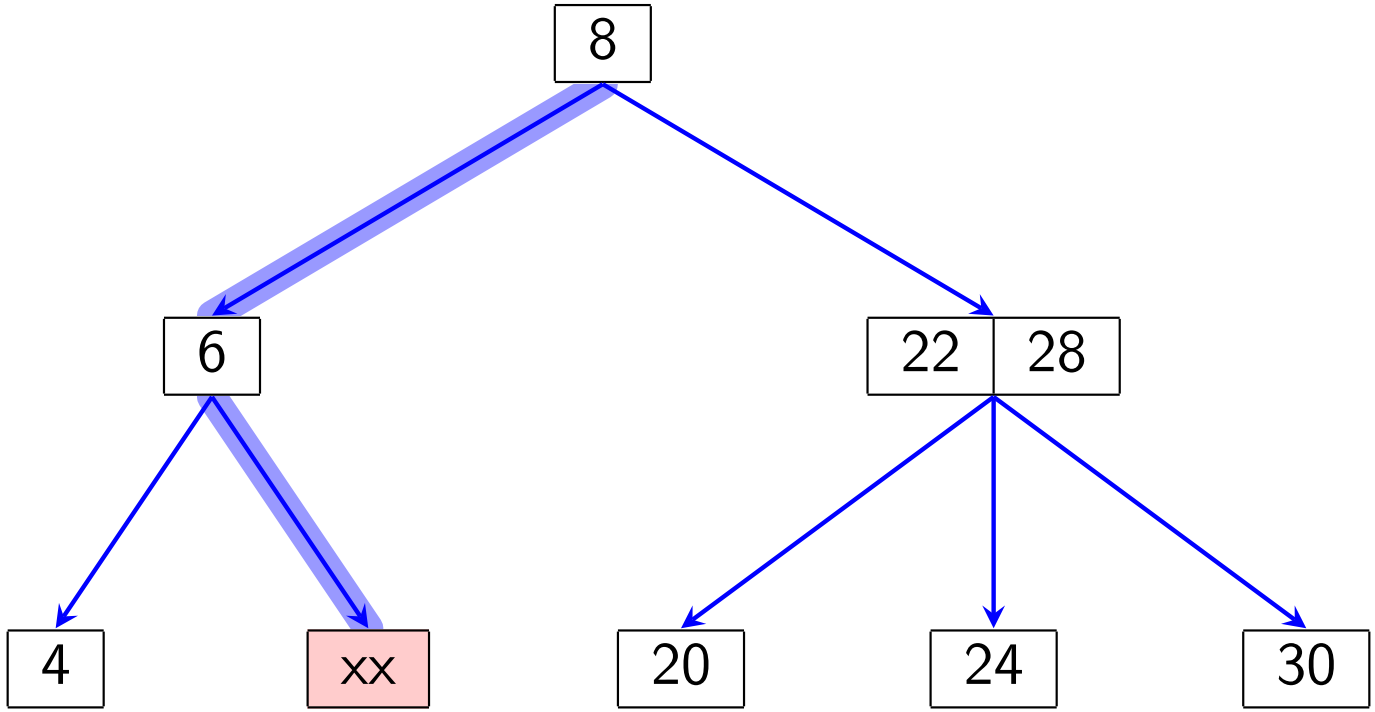
Hình 25: Xóa khóa 26: nút đỏ thiếu khóa

Minh họa xóa phần tử (cont.)



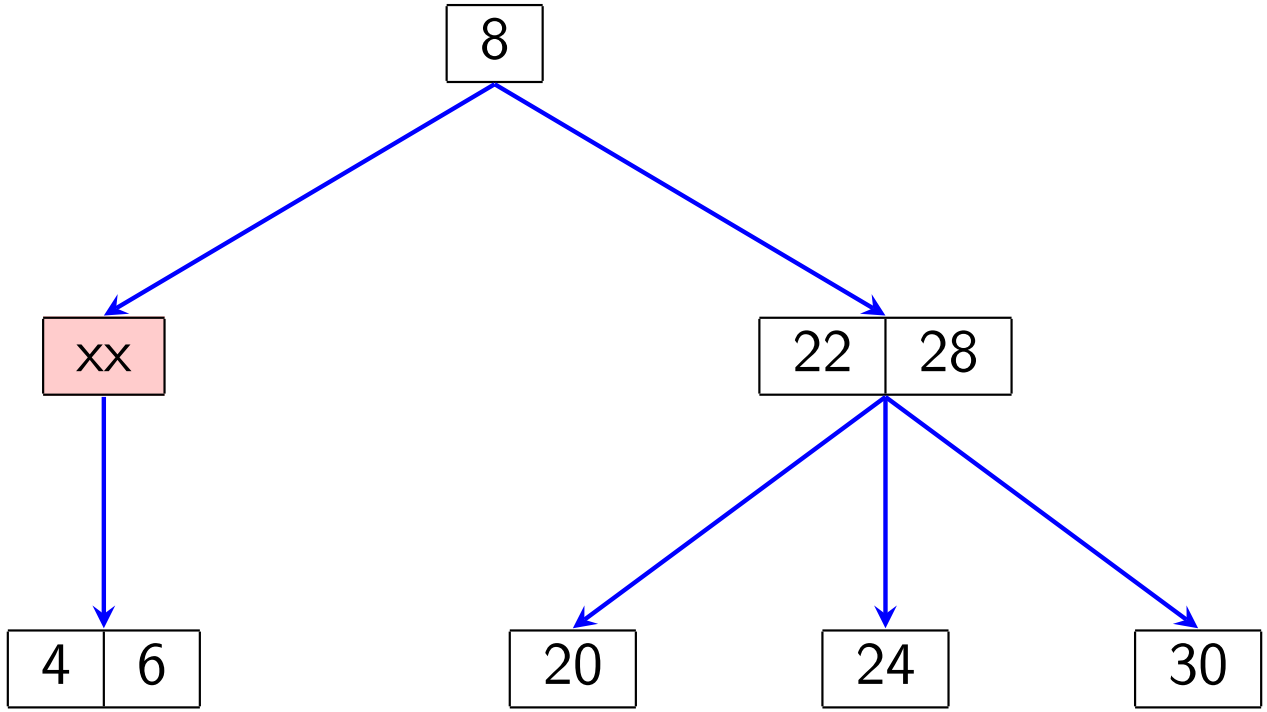
Hình 26: Xóa khóa 26: mượn khóa (trường hợp 1)

Minh họa xóa phần tử (cont.)



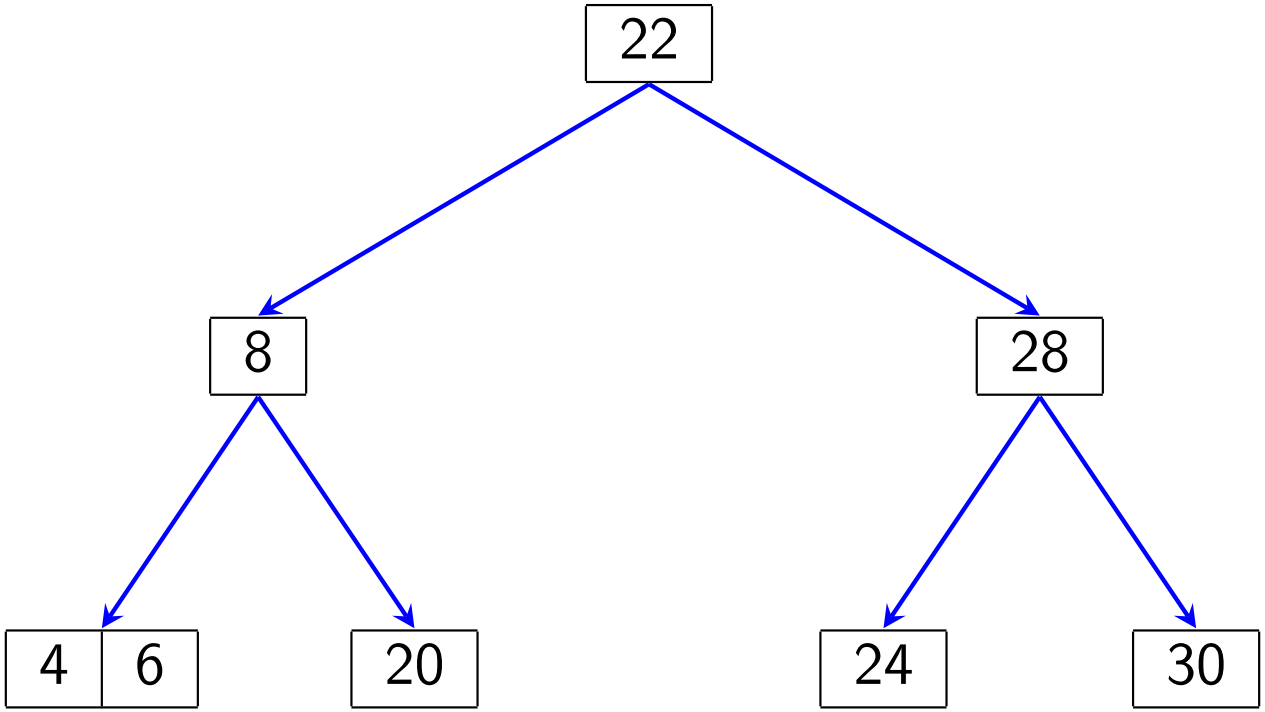
Hình 27: Xóa khóa 18: nút đỏ thiếu khóa

Minh họa xóa phần tử (cont.)



Hình 28: Xóa khóa 18: nhập nút (trường hợp 2), nút đỏ thiếu khóa

Minh họa xóa phần tử (cont.)



Hình 29: Xóa khóa 18: mượn khóa (trường hợp 1)

Bài luyện tập

Ví dụ 2

Hãy xây dựng cây B-cây 3-nhánh từ dãy $\{4, 3, 5, 1, 2, 7, 9, 8, 15, 11, 12, 16\}$

- Xóa các nút 8, 16
- Thêm các nút 6, 17

Định lý về chiều cao của B-cây

Định lý 1

Gọi n $(n \geq 1)$ là số phần tử hay khóa của B-cây m $(m > 2)$ là bậc của cây và h là chiều cao của cây

$$h \leq \log_m \frac{n+1}{2} \tag{4}$$

Đánh giá B-cây

Phân tích chi phí thực hiện theo n (số lượng nút của cây)

	xấu nhất	**trung bình**	**tốt nhất**
tìm một phần tử	?	?	?
thêm một phần tử	?	?	?
xóa một phần tử	?	?	?

Phân tích chi phí bộ nhớ theo n (số lượng nút của cây)

Tài liệu tham khảo