



ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

# **CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

## **CHƯƠNG V**

### **B - TREE**

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, KHU PHỐ 6, PHƯỜNG LINH TRUNG, QUẬN THỦ ĐỨC, TP. HỒ CHÍ MINH

[T] 08 3725 2002 101 | [F] 08 3725 2148 | [W] [www.uit.edu.vn](http://www.uit.edu.vn) | [E] [info@uit.edu.vn](mailto:info@uit.edu.vn)



# **NỘI DUNG**

- I. Khái niệm B - Tree**
- II. Các thao tác trên B – Tree**
  - 1. Thêm node**
  - 2. Xóa node**



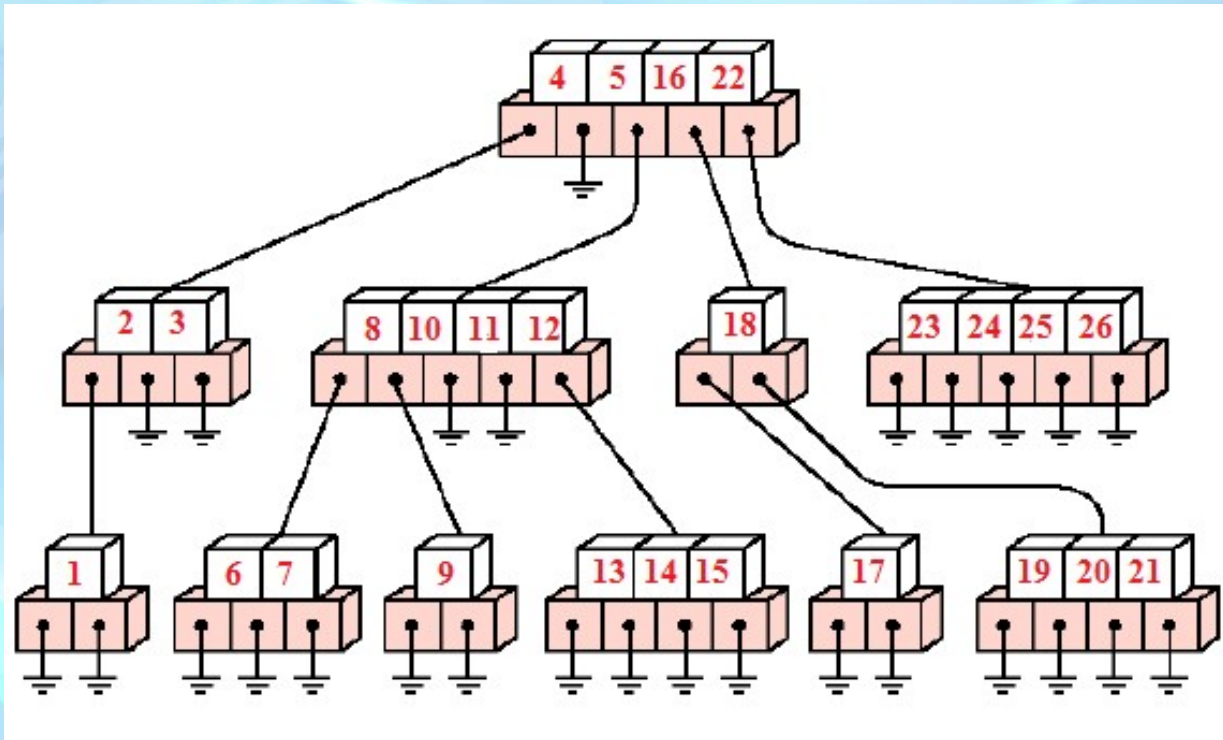
# I. Khái niệm B - Tree

- ❖ B – Tree là cấu trúc dữ liệu phù hợp cho việc lưu trữ ngoài do R.Bayer và E.M.McCreight đưa ra năm 1972
- ❖ B – Tree là một cây nhiều nhánh tìm kiếm nên nó sẽ mang các tính chất của cây nhiều nhánh tìm kiếm



# I. Khái niệm B - Tree

- ❖ Cây nhiều nhánh tìm kiếm bậc m
  - Mỗi node có tối đa m cây con và m-1 khóa
  - Ở tại mỗi node các khóa được sắp xếp theo thứ tự tăng dần
  - Giá trị các khóa ở mỗi node con sẽ tuân theo qui tắc như hình







# I. Khái niệm B - Tree

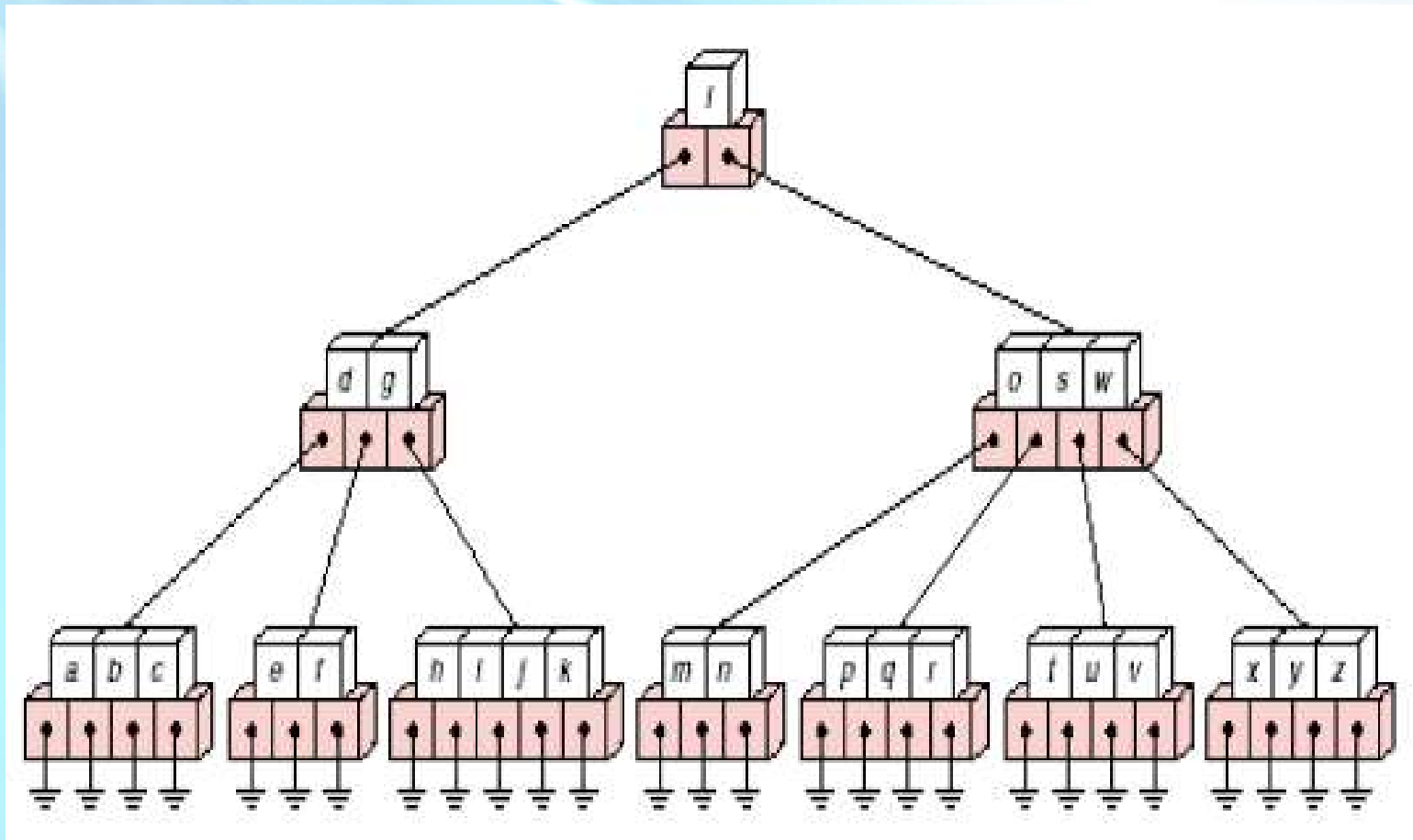
Cho số tự nhiên  $k > 0$ , B-Trees bậc  $m$  với  $m = 2*k+1$  là một cây thỏa mãn các tính chất:

- Tất cả node lá nằm trên cùng một mức
- Tất cả các node, trừ node gốc và node lá, có **\*tối thiểu\***  $k+1$  node con.
- Tất cả các node có **\*tối đa\***  $m$  con
- Tất cả các node, trừ node gốc, có từ  $k$  cho đến  $m-1$  khóa (keys). Node gốc có từ 1 đến  $m-1$  khóa.
- Một node không phải lá và có  $n$  khóa thì phải có  $n+1$  node con.



# I. Khái niệm B - Tree

## ❖ Cây B – Tree bậc 5 có 3 mức





## II. Các thao tác trên B - Tree

- ❖ B – Tree là cây tự cân bằng (self-balancing), nghĩa là khi thêm hoặc xoá 1 node thì cây sẽ có những action để đảm bảo chiều cao của cây càng thấp càng tốt.



## II. Các thao tác trên B - Tree

- ❖ Thêm node tuân thủ theo quy tắc sau:
  - Thêm node theo thứ tự tăng dần từ trái qua phải
  - Nếu bị **tràn** thì **tách node** → chọn node ở giữa làm node cha
  - Ví dụ: Thêm các node 5,7,3,6,9 vào cây B-Tree bậc 5





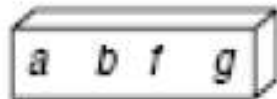


## II. Thêm node trên B - Tree

- ❖ Tạo cây B – Tree bậc 5 từ dãy các khóa sau: a, g, f, b, k, d, h, m, i, e, s, i, r, x, c, l, n, t, u, p

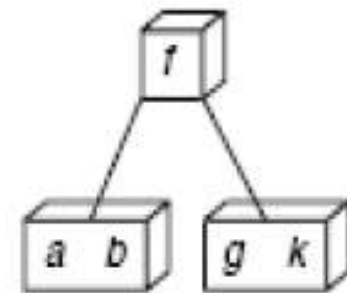
1.

a, g, f, b:



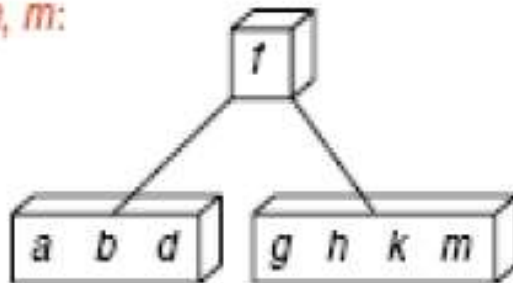
2.

k:



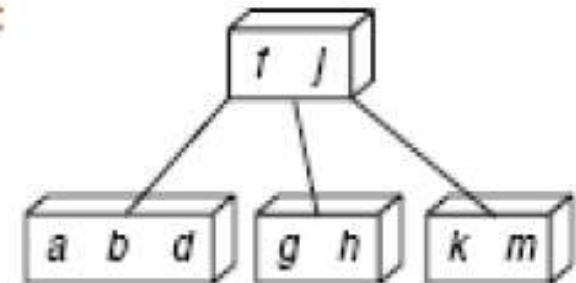
3.

d, h, m:



4.

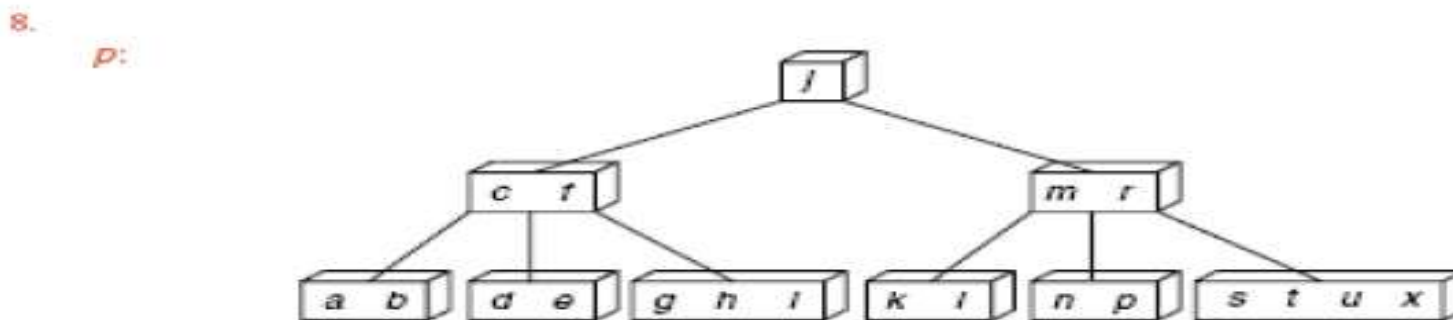
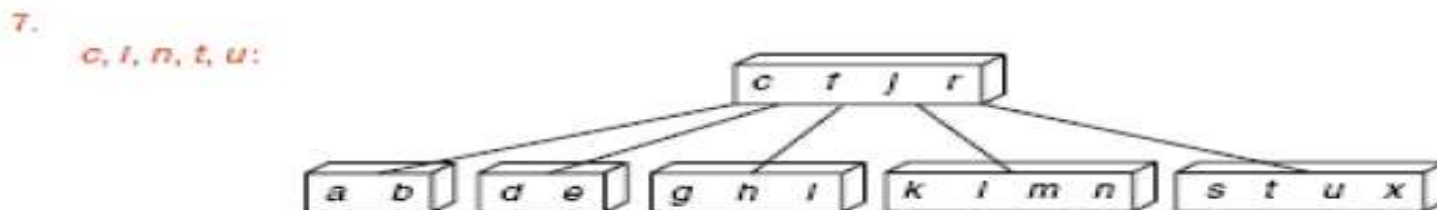
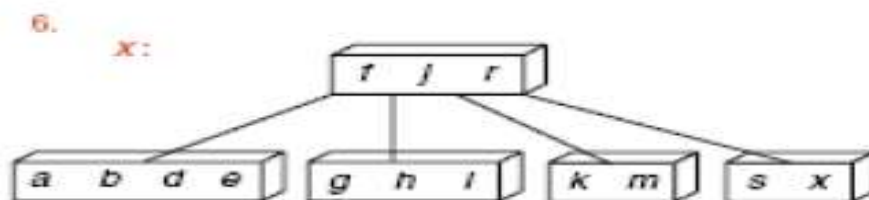
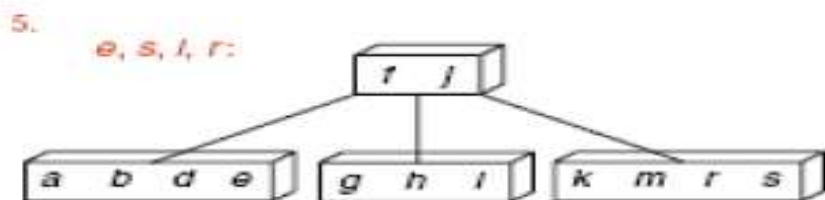
j:





## II. Thêm node trên B - Tree

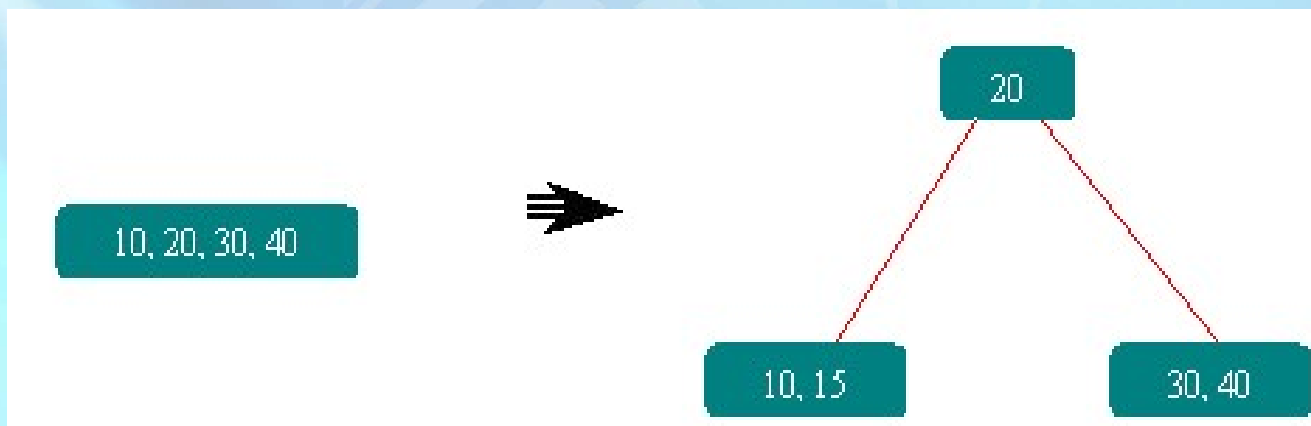
- ❖ Tạo cây B – Tree bậc 5 từ dãy các khóa sau: a, g, f, b, k, d, h, m, i, e, s, i, r, x, c, l, n, t, u, p





## II. Thêm node trên B - Tree

- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





## II. Thêm node trên B - Tree

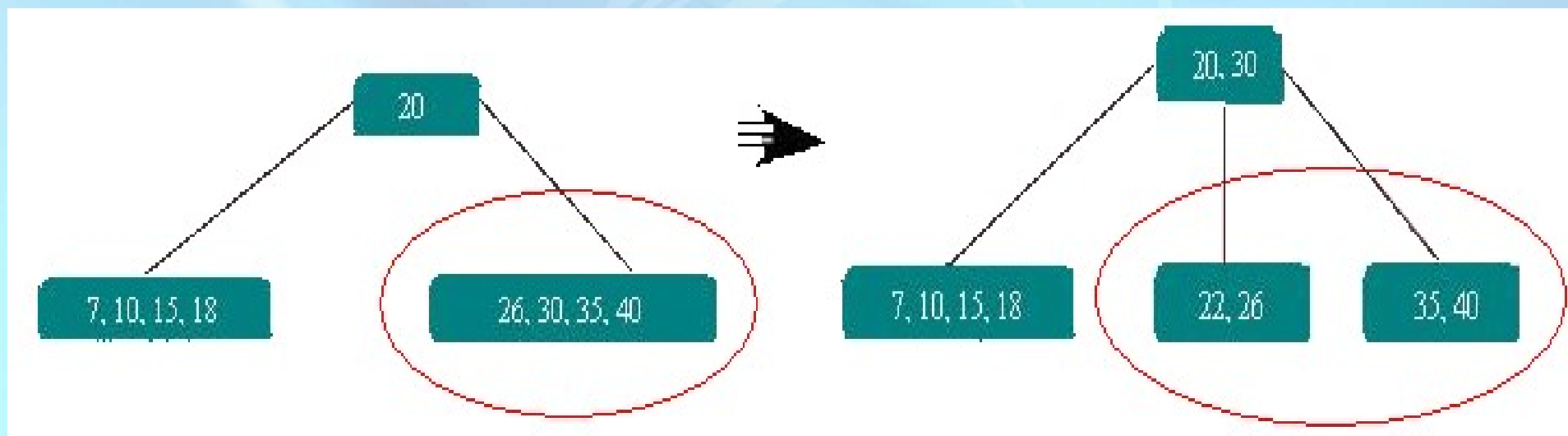
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





## II. Thêm node trên B - Tree

- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;







## II. Thêm node trên B - Tree

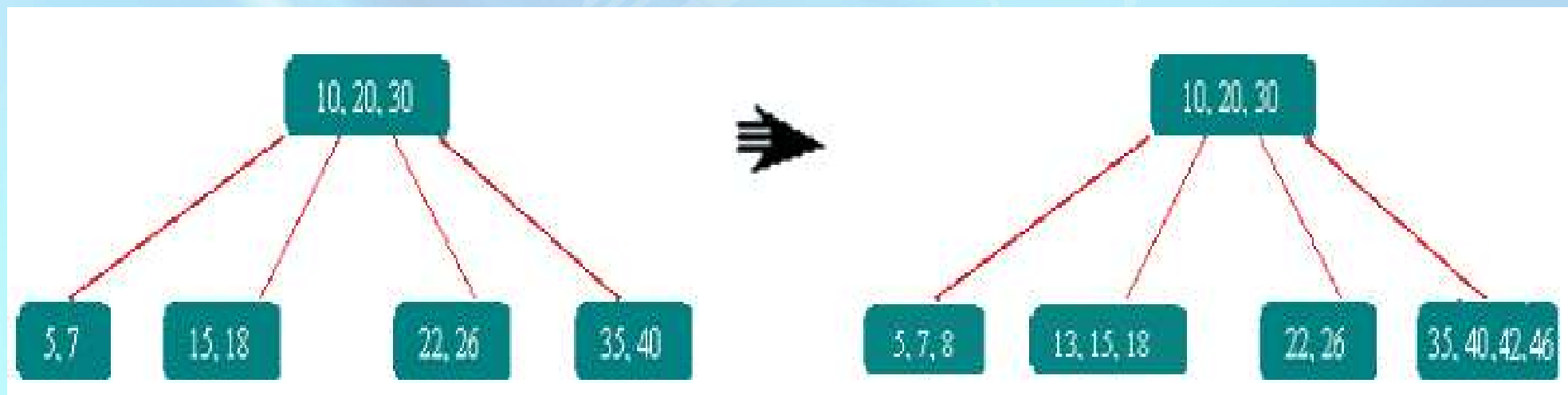
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





## II. Thêm node trên B - Tree

- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





## II. Thêm node trên B - Tree

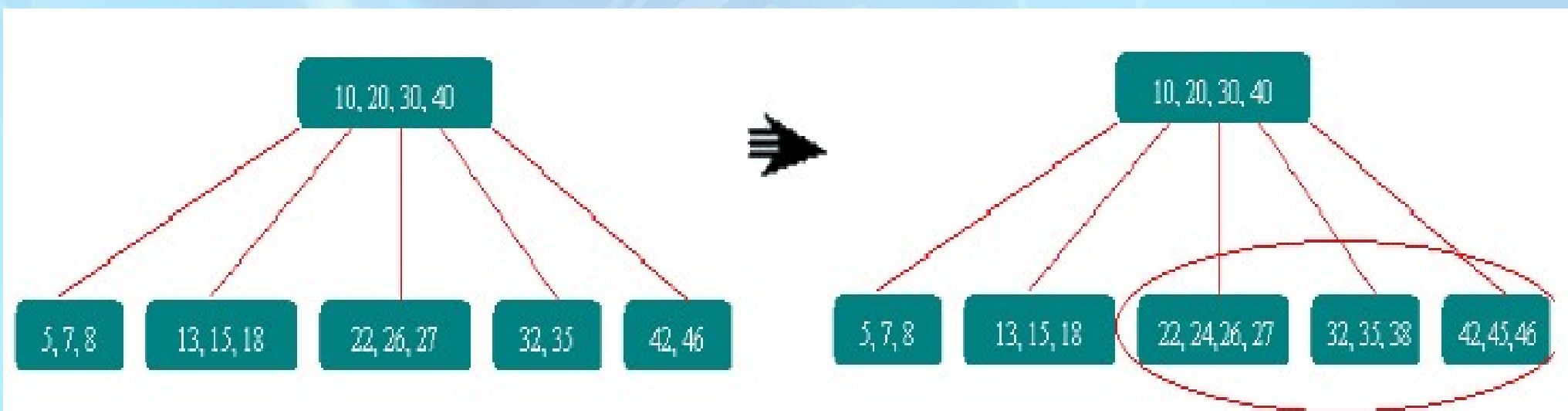
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





## II. Thêm node trên B - Tree

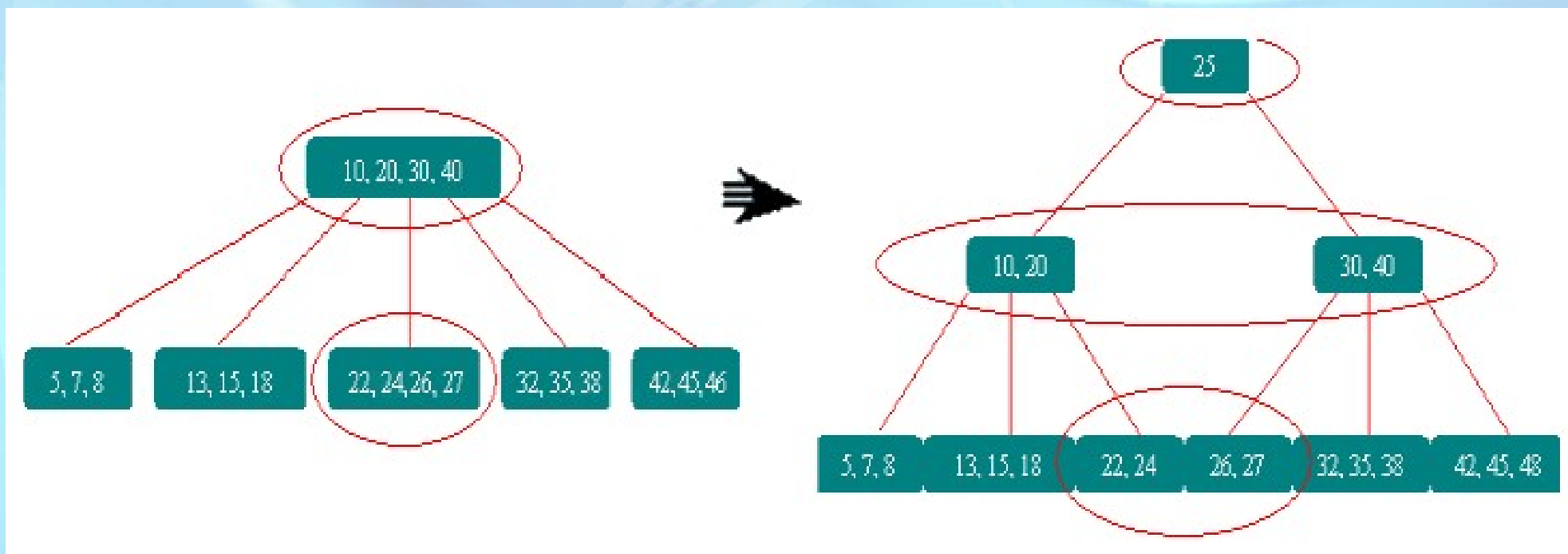
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





## II. Thêm node trên B - Tree

- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;







## II. Các thao tác trên B - Tree

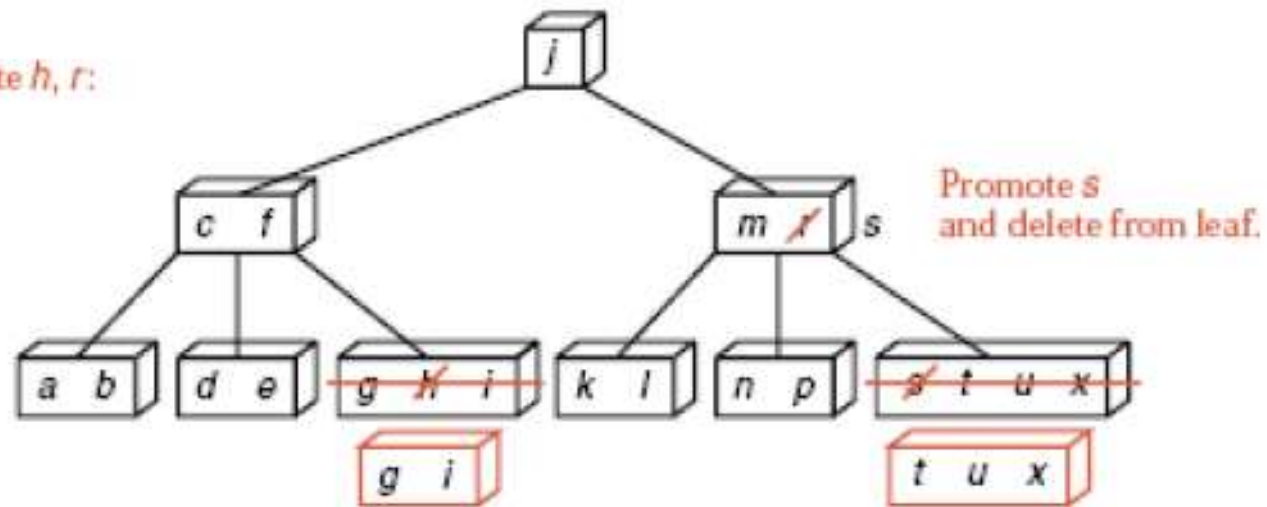
- ❖ Xóa node tuân thủ theo qui tắc sau:
  - Xóa node lá phải đảm bảo nguyên tắc số cây con **\*tối thiểu\***. Nếu xóa node lá mà làm node thiếu cây con thì lấy node cha trực tiếp của node bị xóa làm node thay thế vào vị trí node đã xóa.
  - Xóa node không phải node lá thì lấy node con trực tiếp của nó lên thay thế vị trí node đã xóa nếu việc đó không làm ảnh hưởng điều kiện tối thiểu. Nếu không được thì combine đa tầng.



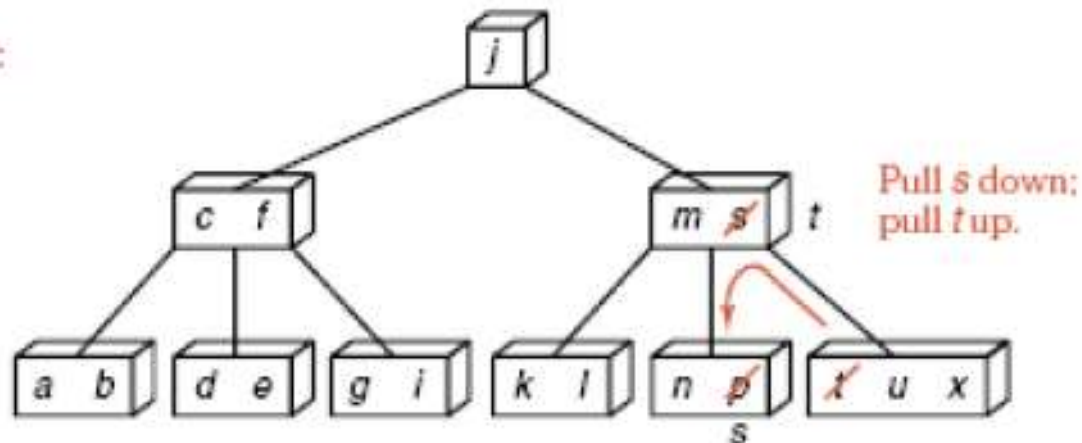
## II. Xóa node trên B - Tree



1. Delete  $h, r$ :



2. Delete  $p$ :



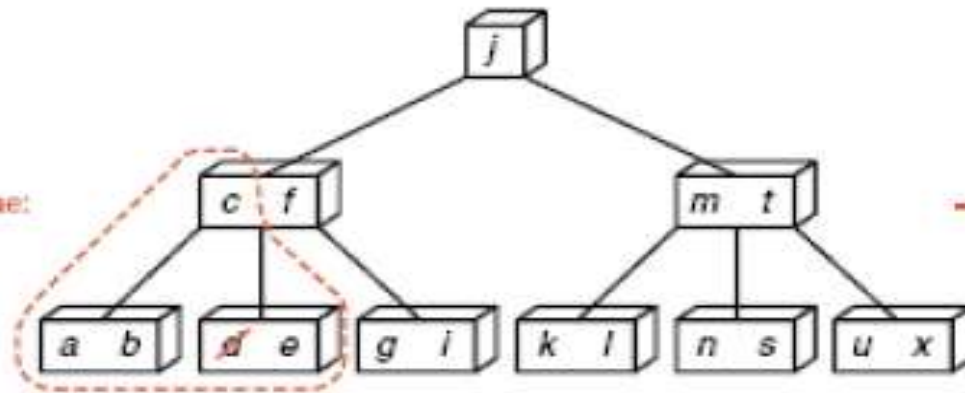


## II. Xóa node trên B - Tree

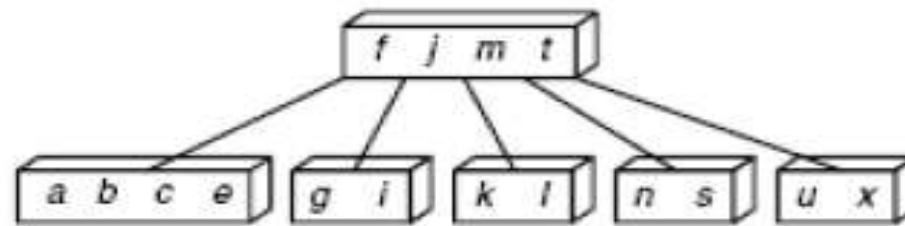
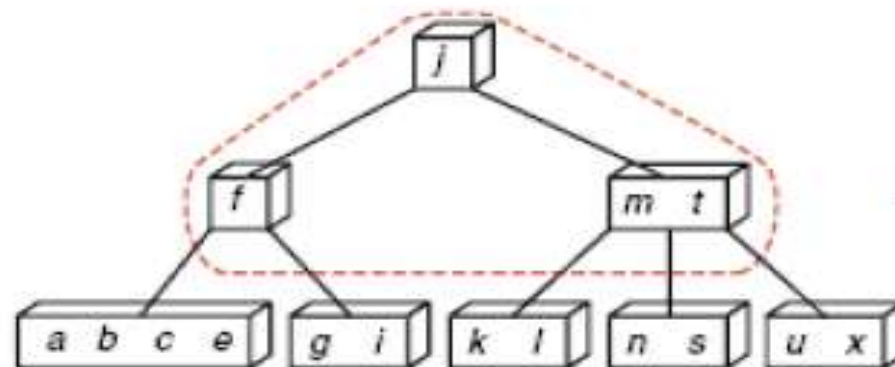


3. Delete  $d$ :

Combine:

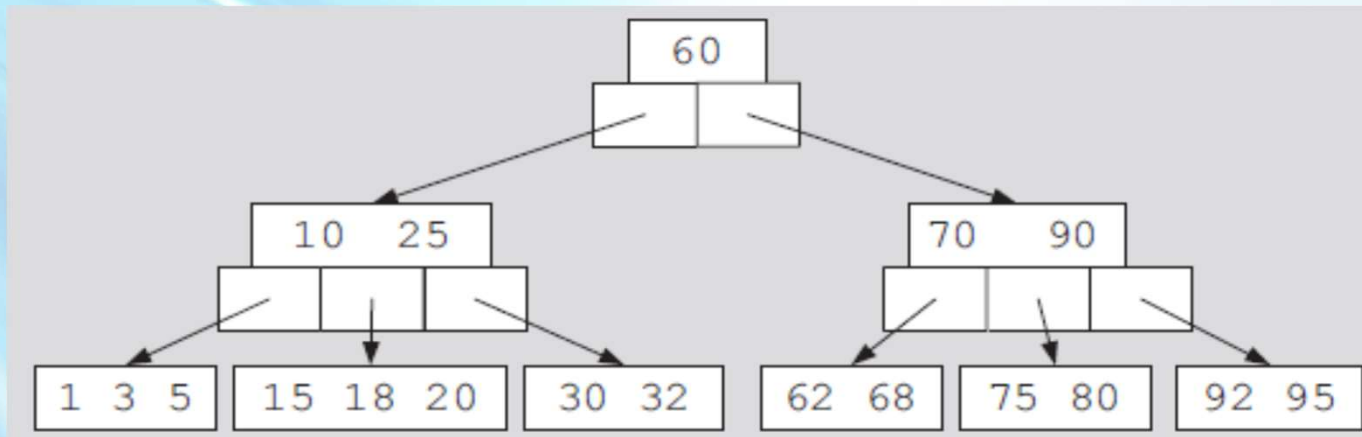


Combine:

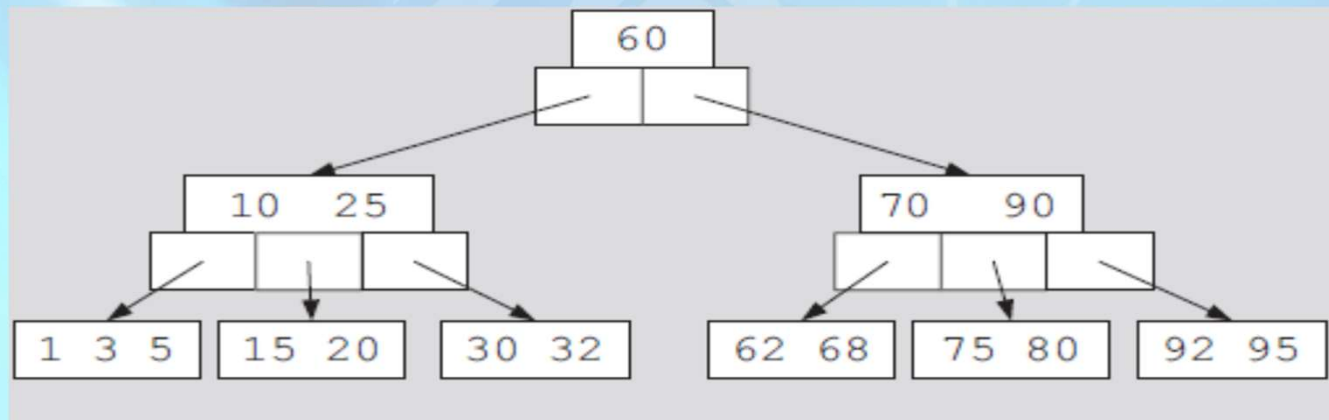




## II. Xóa node trên B - Tree



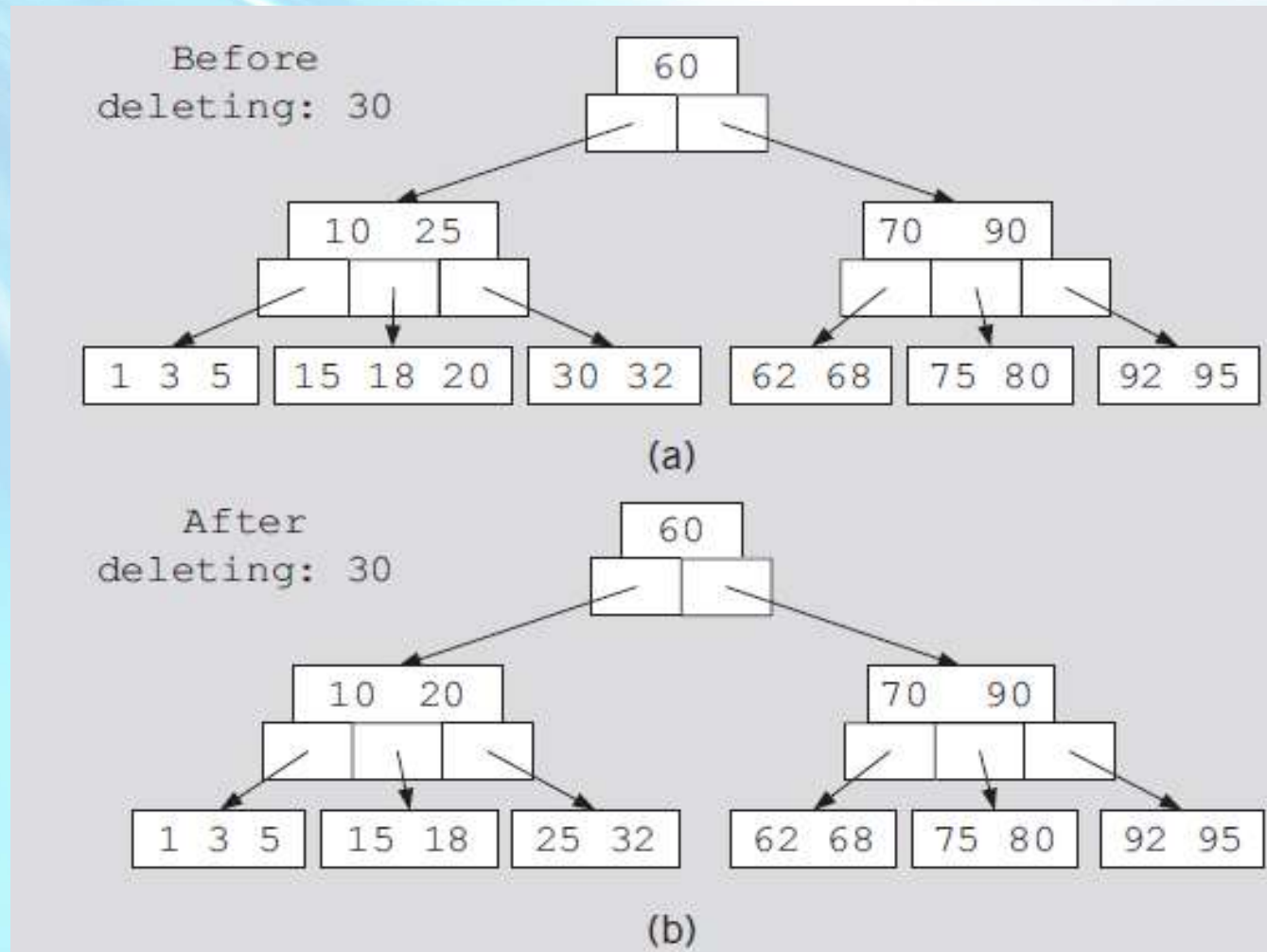
**FIGURE 11-30** A B-tree of order 5



**FIGURE 11-31** Deleting 18 from  
a B-tree of order 5



## II. Xóa node trên B - Tree

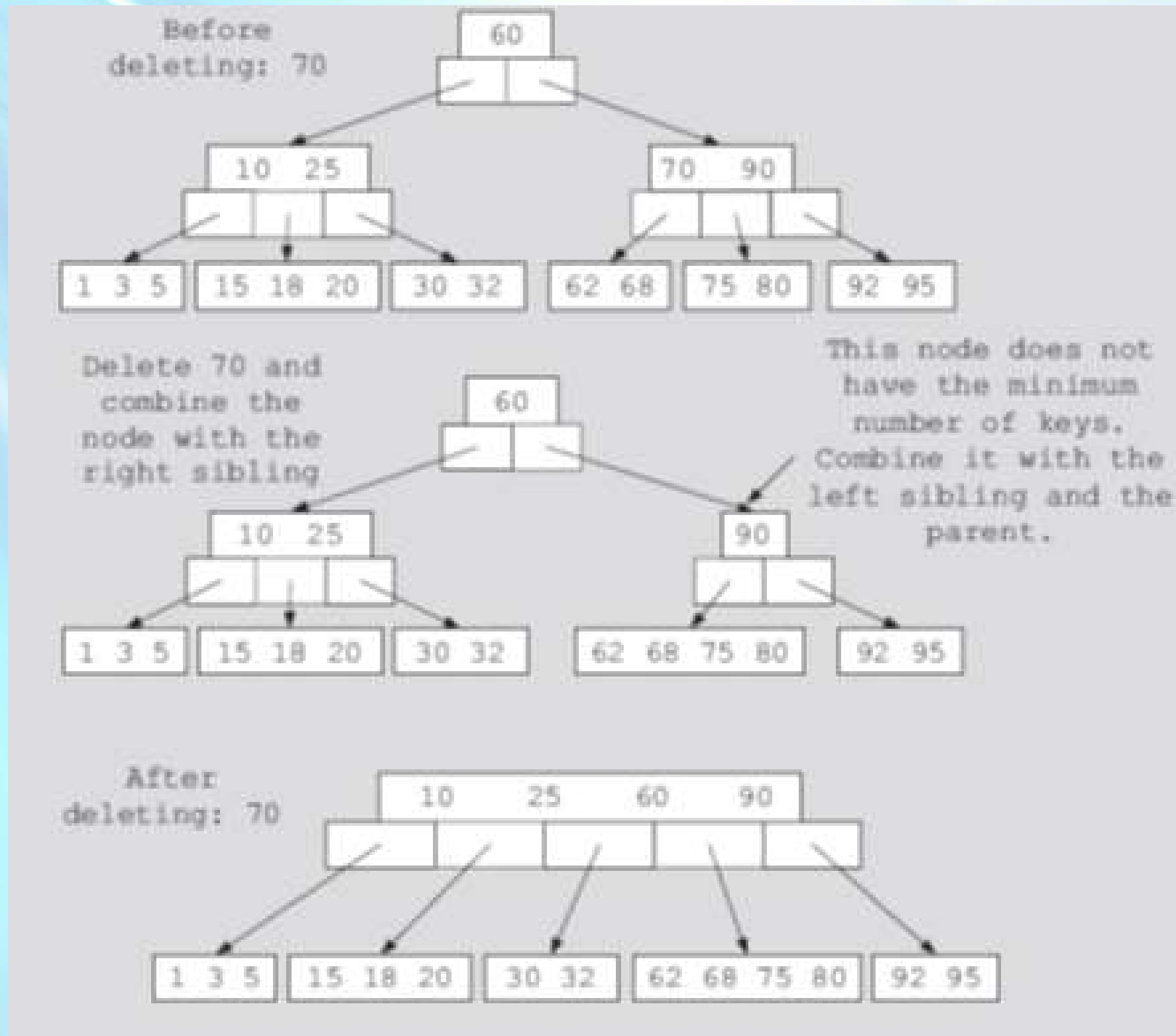


**FIGURE 11-32** B-tree before and after deleting 30





## II. Xóa node trên B - Tree





# III. Bài tập

- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau: 5, 30, 40, 70, 16, 82, 95, 100, 73, 54, 98, 37, 25, 62, 81, 150, 79, 181
- ❖ Xóa node 30, 54, 70, 25