



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



NỘI DUNG

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



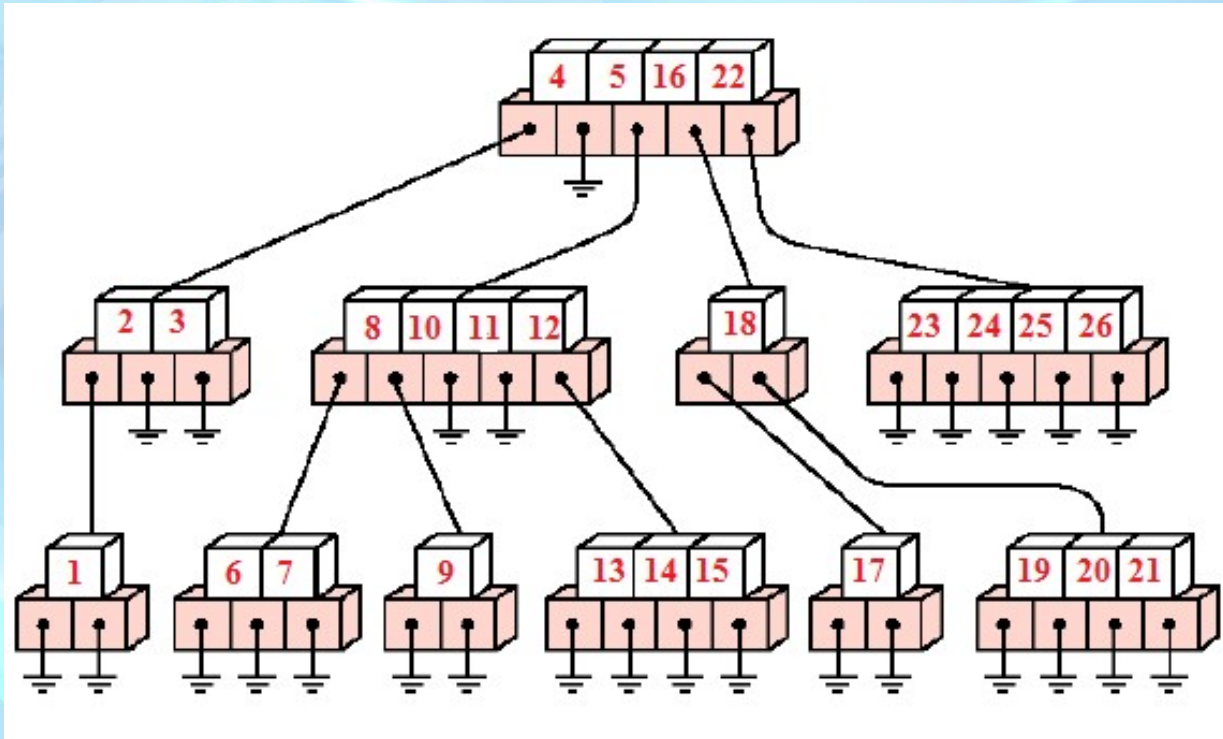
I. Khái niệm B - Tree

- ❖ 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) có ***tối thiểu*** $k+1$ node con (k khóa)
- 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

Đối với cây B-Trees bậc m với $m = 2*k+1$ thì lưu ý điều kiện tối đa và tối thiểu

- Tối đa: $2*k$ khóa / node
- Tối thiểu: k khóa / node (trừ node gốc)

Ví dụ: cây B-Tree bậc 5

- Tối đa: 4 khóa / node
- Tối thiểu: 2 khóa / node (trừ node gốc)

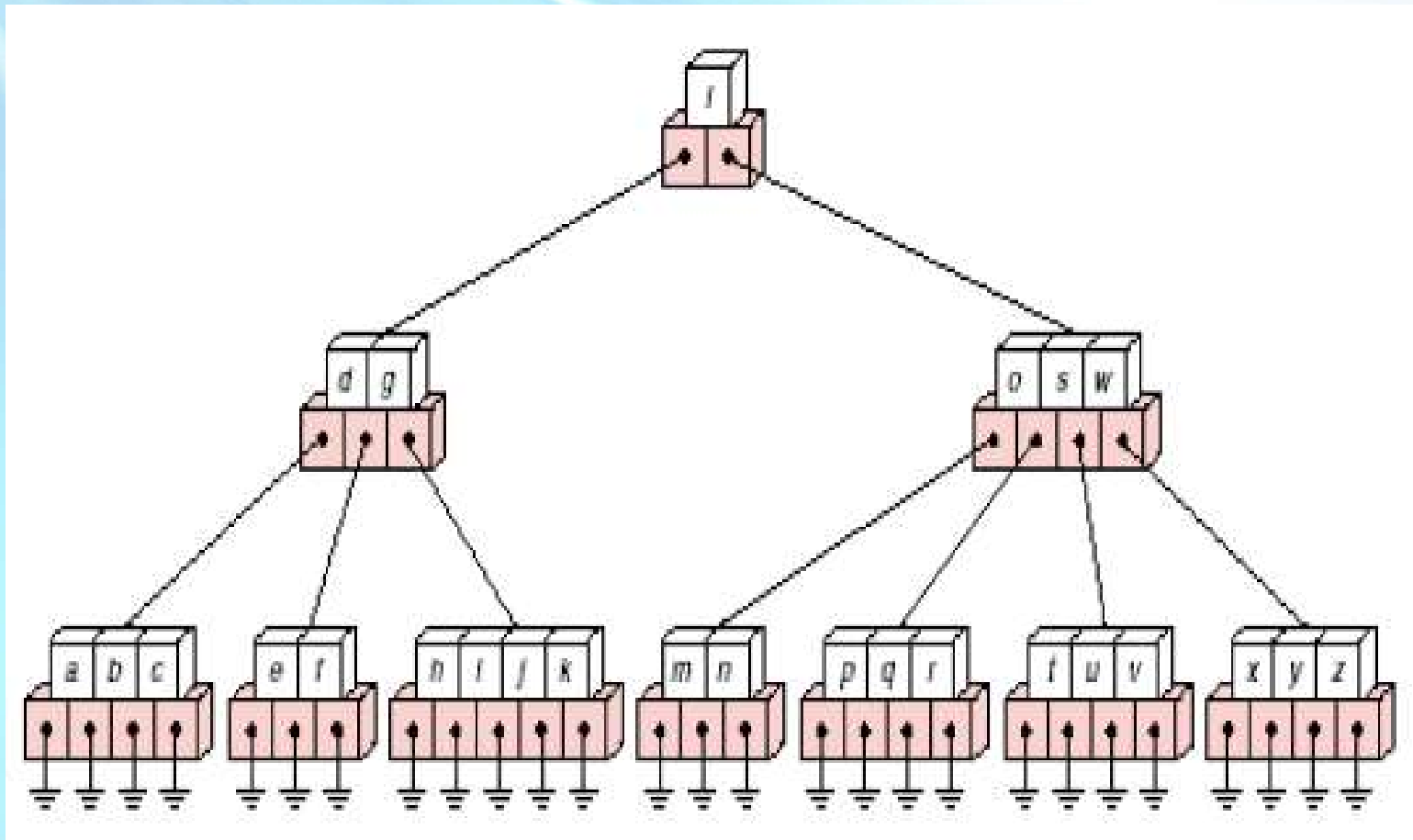
Ví dụ: cây B-Tree bậc 3

- Tối đa: 2 khóa / node
- Tối thiểu: 1 khóa / node (trừ node gốc)



I. Khái niệm B - Tree

❖ Cây B – Tree bậc 5 có 3 mức





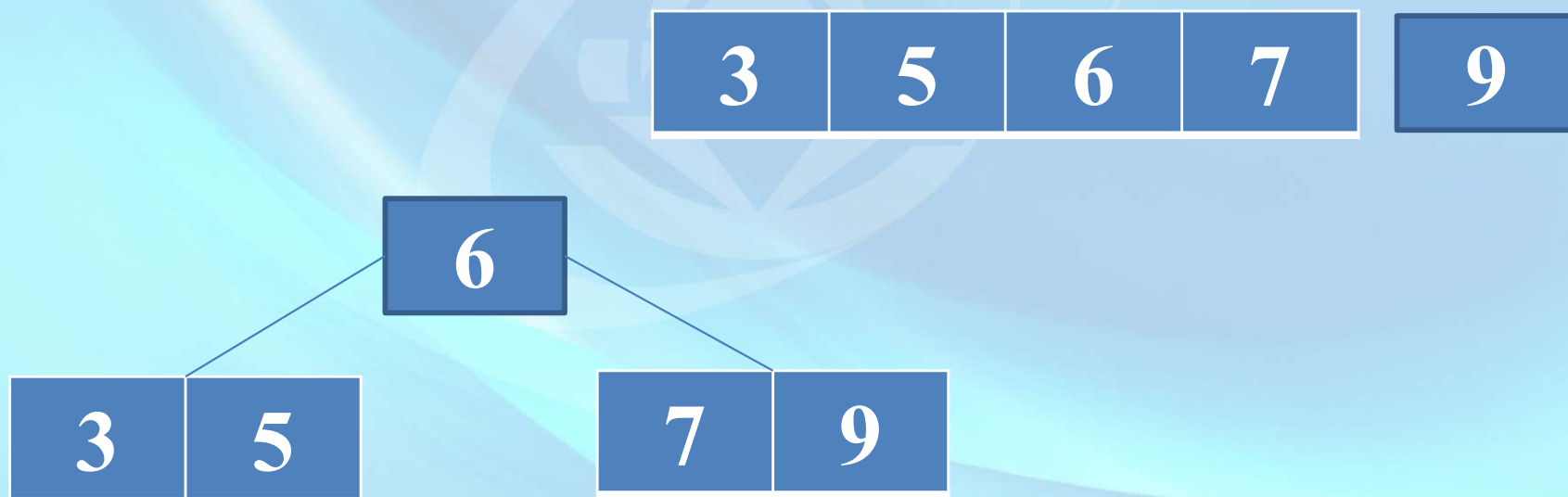
II. Các thao tác trên B - Tree

- ❖ 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 khóa tuân thủ theo quy tắc sau:
 - Thêm khóa cần sắp xếp theo thứ tự tăng dần từ trái qua phải
 - Nếu khóa cần thêm vào làm vi phạm điều kiện tối đa thì **tách (slipt) node** → chọn khóa ở giữa làm node cha
 - Ví dụ: Thêm các khóa 5,7,3,6,9 vào cây B-Tree bậc 5



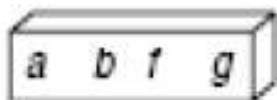


II. Thêm node trên B - Tree

- ❖ 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

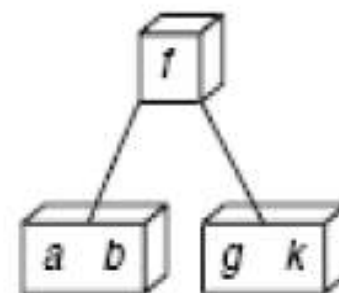
1.

a, g, f, b:



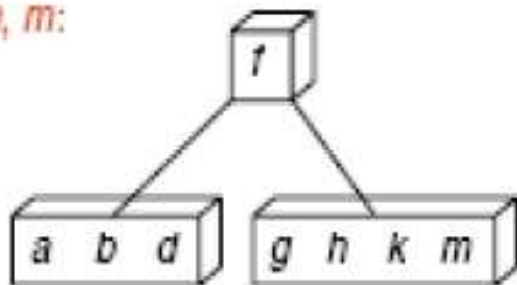
2.

k:



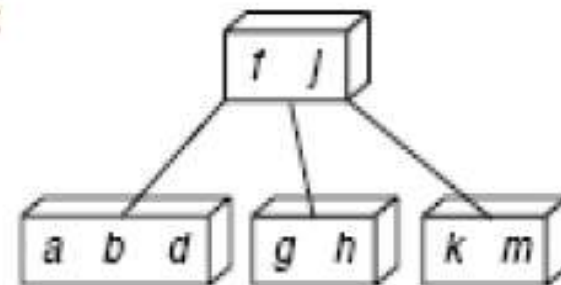
3.

d, h, m:



4.

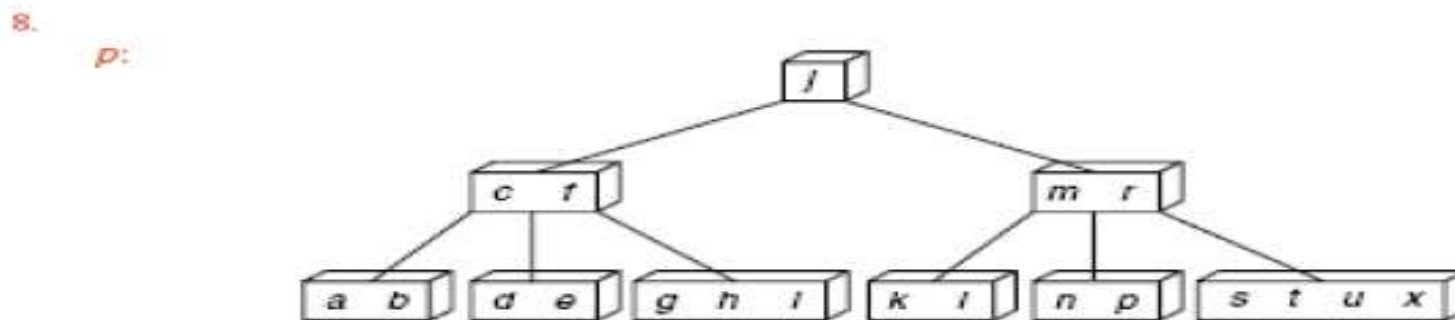
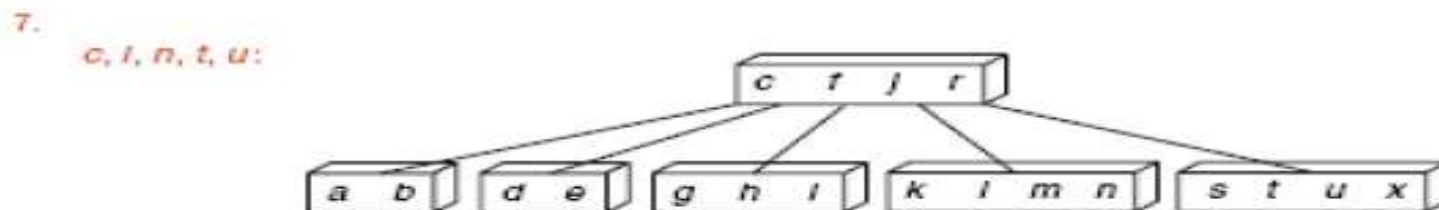
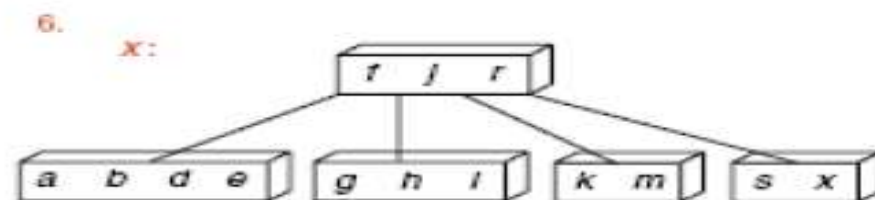
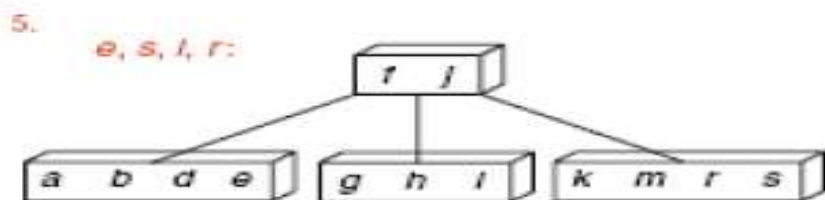
j:





II. Thêm node trên B - Tree

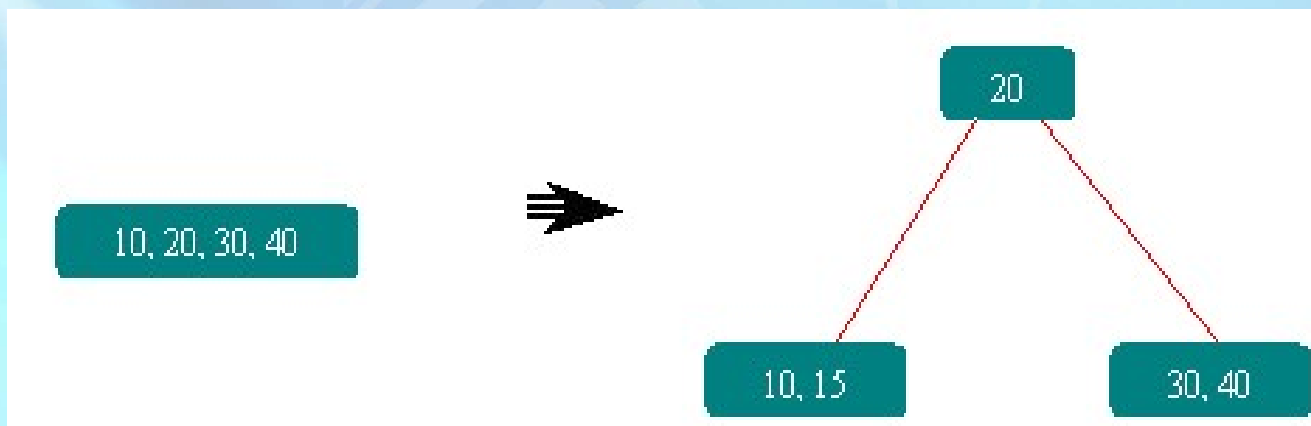
- ❖ Tạo cây B – Tree bậc 5 từ dãy các khóa sau: a, g, f, b, k, d, h, m, i, e, s, i, r, x, c, l, n, t, u, p





II. Thêm node trên B - Tree

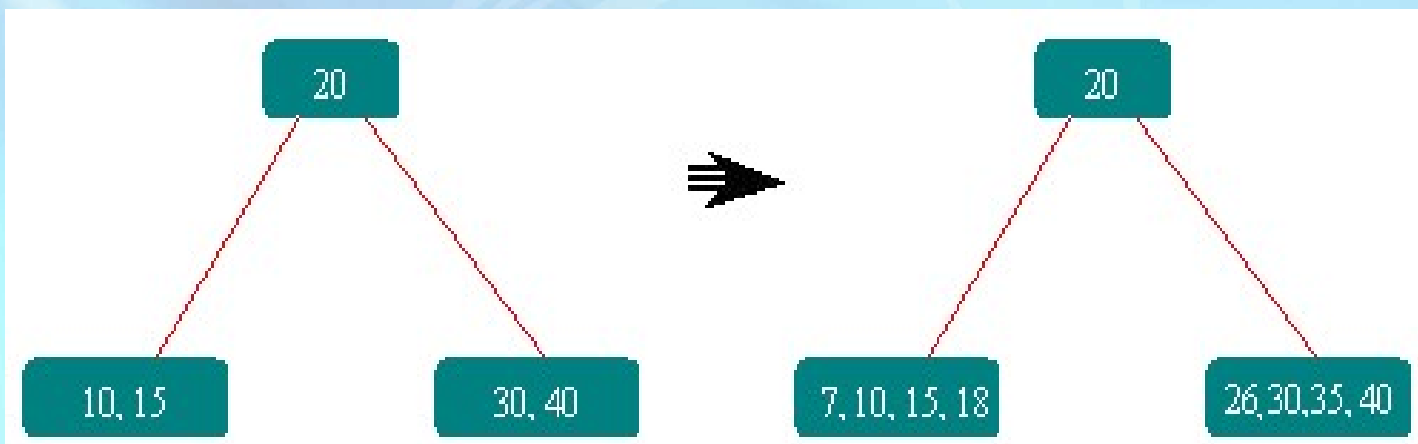
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





II. Thêm node trên B - Tree

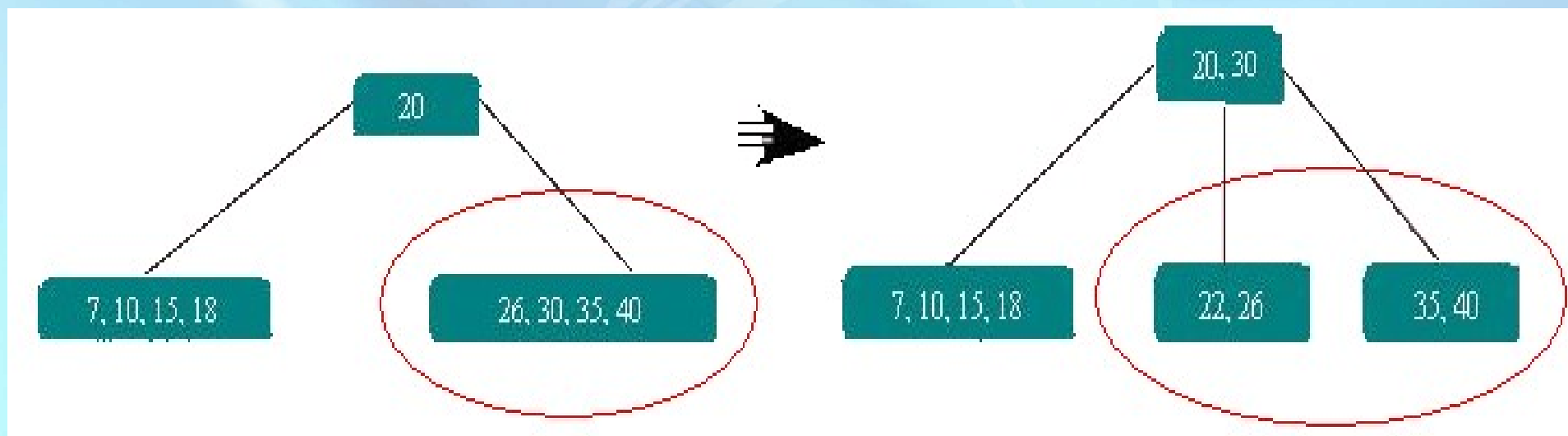
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





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

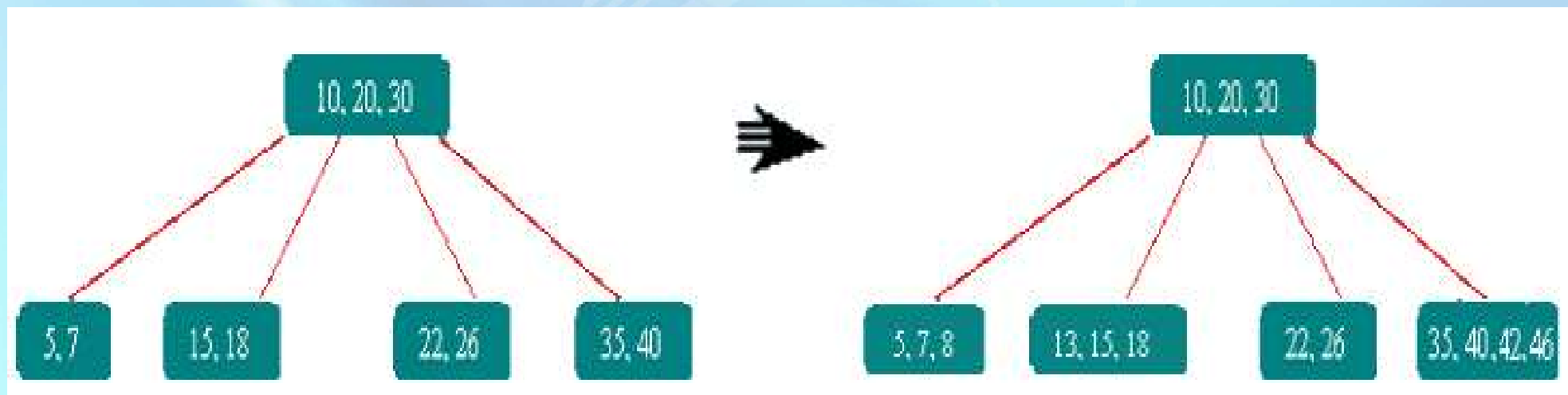
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





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

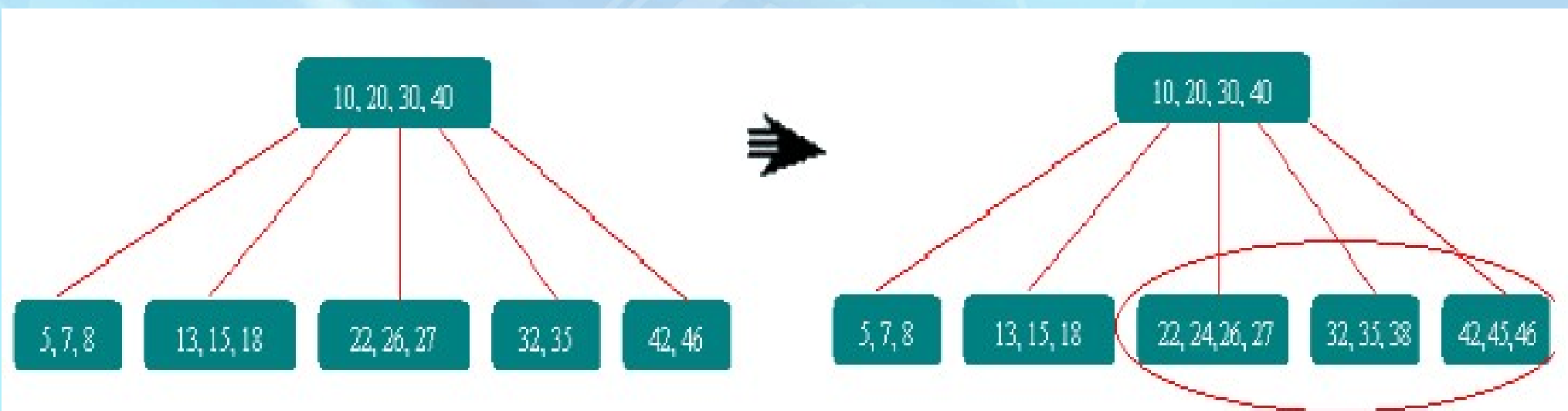
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





II. Thêm node trên B - Tree

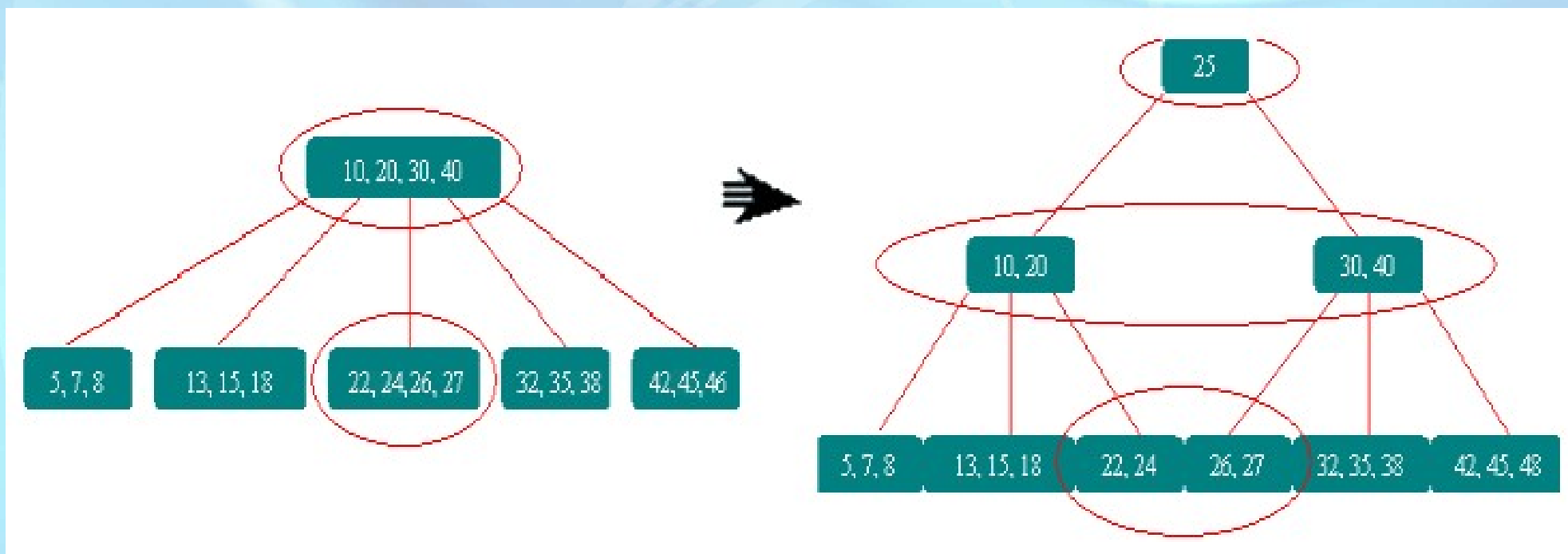
- ❖ Tạo cây B-Tree bậc 5 từ dãy các khóa sau : 20; 40; 10; 30; 15; 35; 7; 26; 18; 22; 5; 42; 13; 46; 27; 8; 32; 38; 24; 45; 25;





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 khóa có các trường hợp như sau:

- ❖ Xóa khóa ở node lá không làm vi phạm đk tối thiểu
- ❖ Xóa khóa ở node lá làm vi phạm đk tối thiểu
 - ❖ **Underflow**: khi xóa khóa làm node vi phạm mà node kế bên có số lượng khóa nhiều hơn tối thiểu
 - mượn khóa thay thế --> không làm thay đổi số khóa của node cha
 - ❖ **Catenation**: khi xóa khóa làm node vi phạm mà 2 node kế nhau có số lượng khóa $< 2*k$
 - gộp node --> làm node cha bị mất khóa (ưu tiên chọn node liền trước để gộp)

Lưu ý ưu tiên chọn **Underflow** hơn **Catenation**

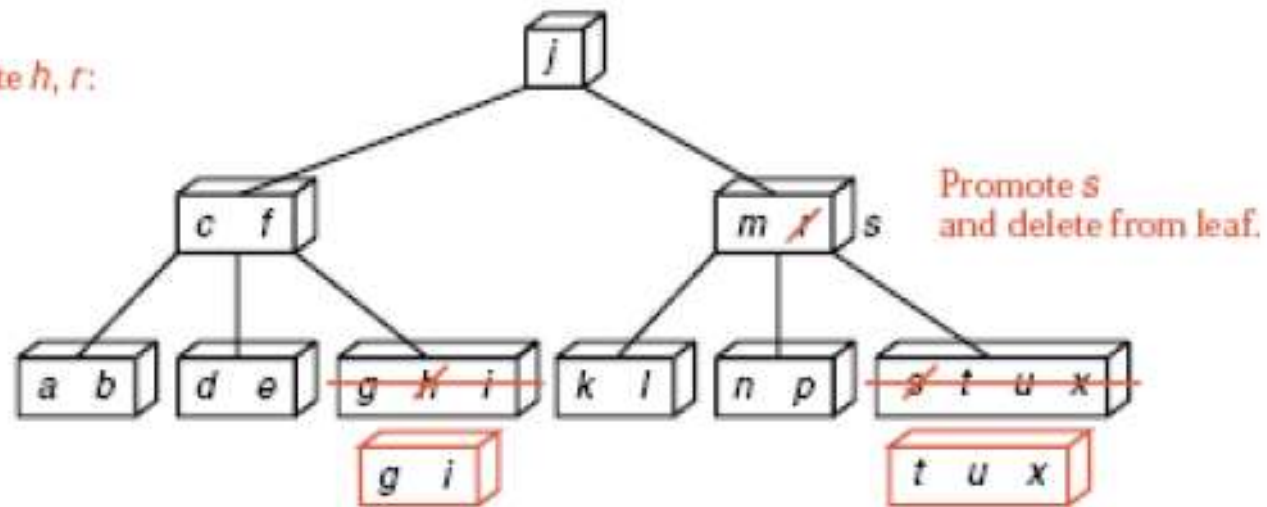
- ❖ Xóa khóa ở node khác lá --> tìm khóa ở node lá thay thế (**max_left**, min_right) bằng cách hoán vị node cần xóa với node lá thay thế --> xử lý giống trường hợp xóa node lá ở trên.



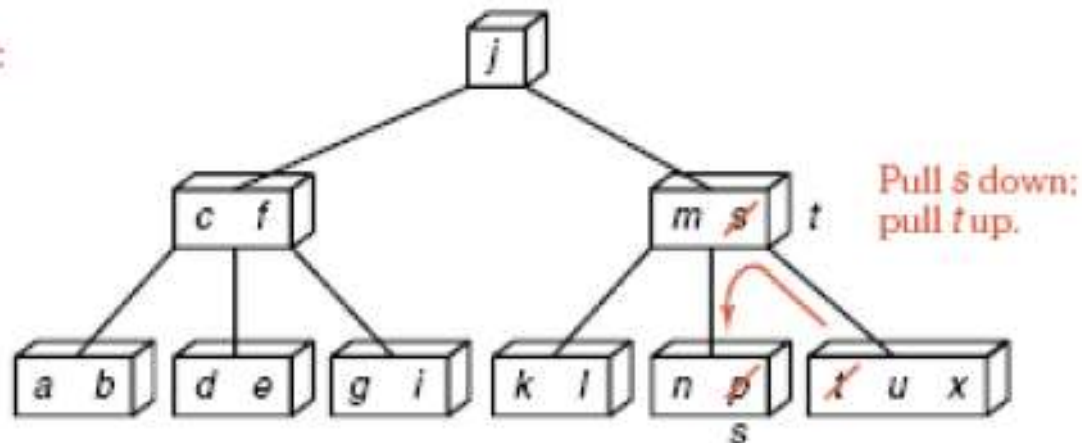
II. Xóa node trên B - Tree



1. Delete h, r :



2. Delete p :



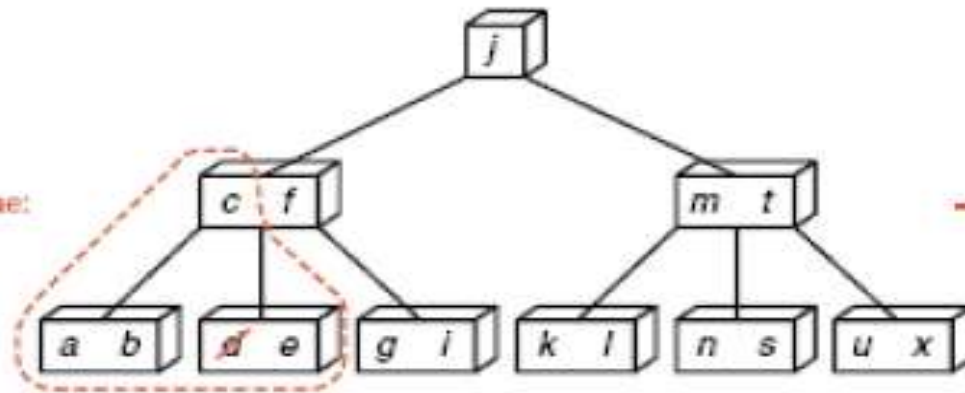


II. Xóa node trên B - Tree

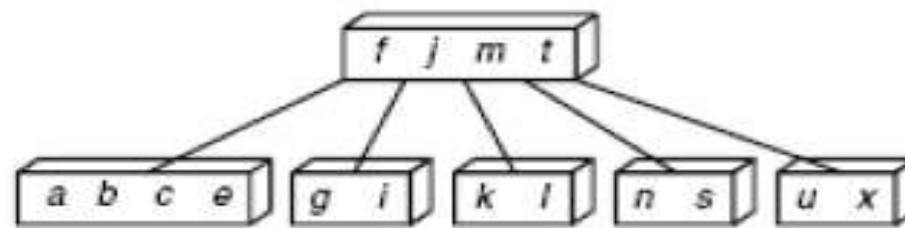
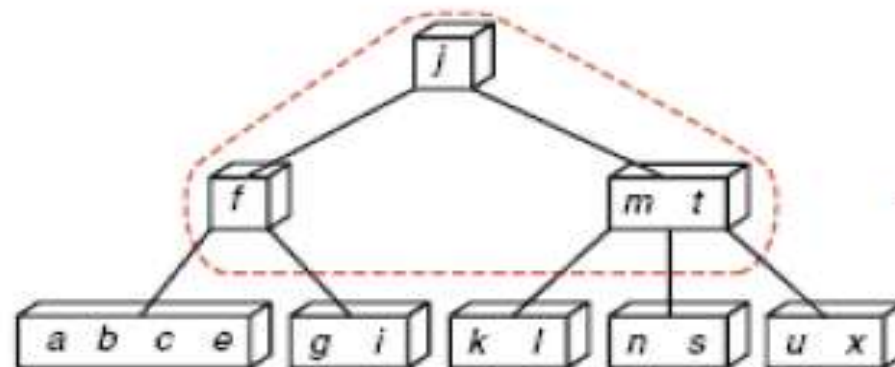


3. Delete d :

Combine:



Combine:





II. Xóa node trên B - Tree

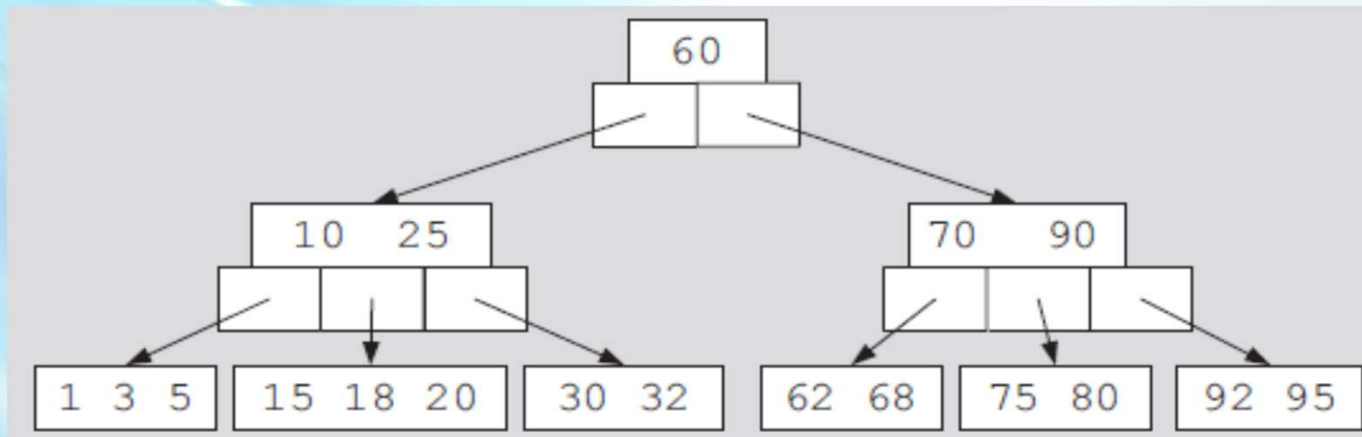


FIGURE 11-30 A B-tree of order 5

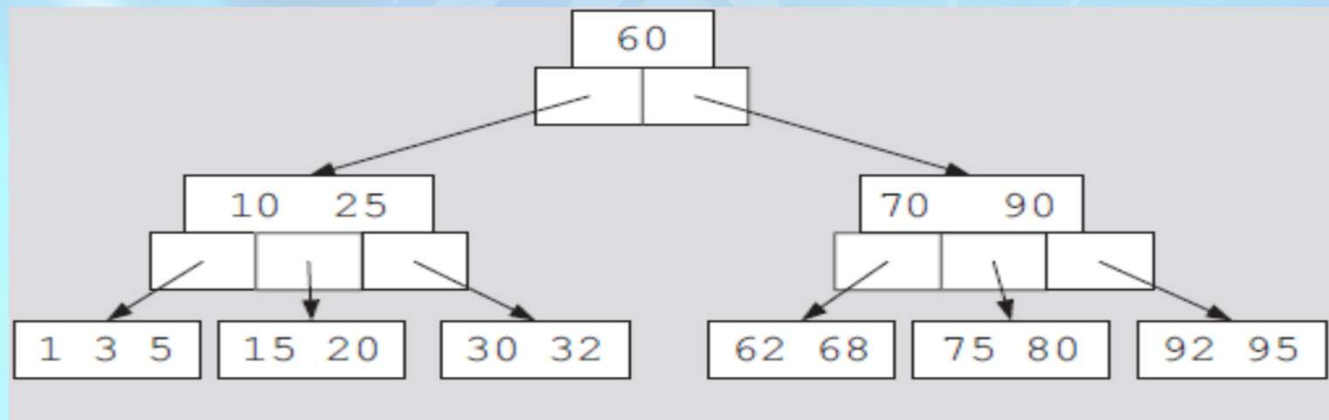


FIGURE 11-31 Deleting 18 from
a B-tree of order 5



II. Xóa node trên B - Tree

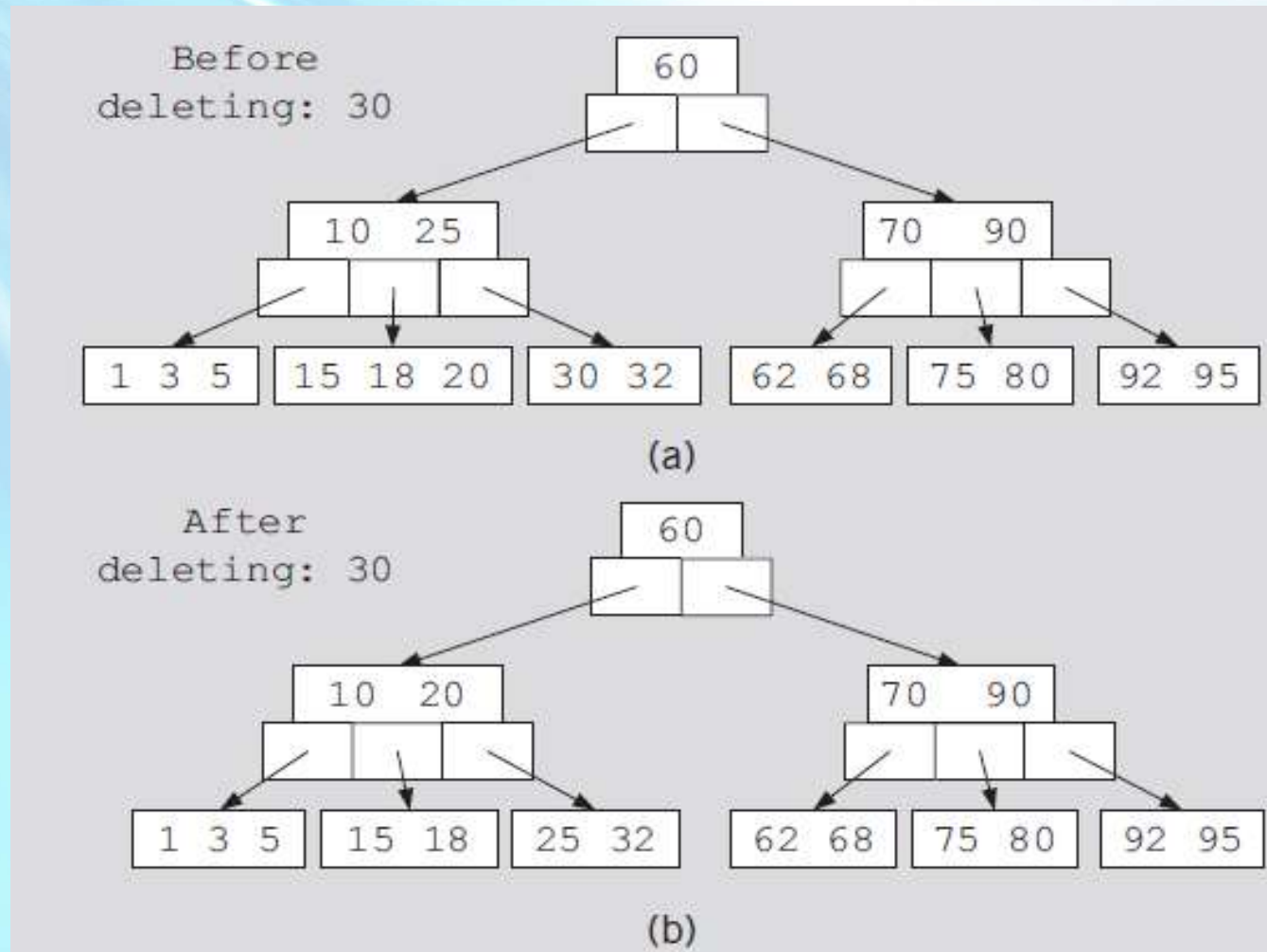
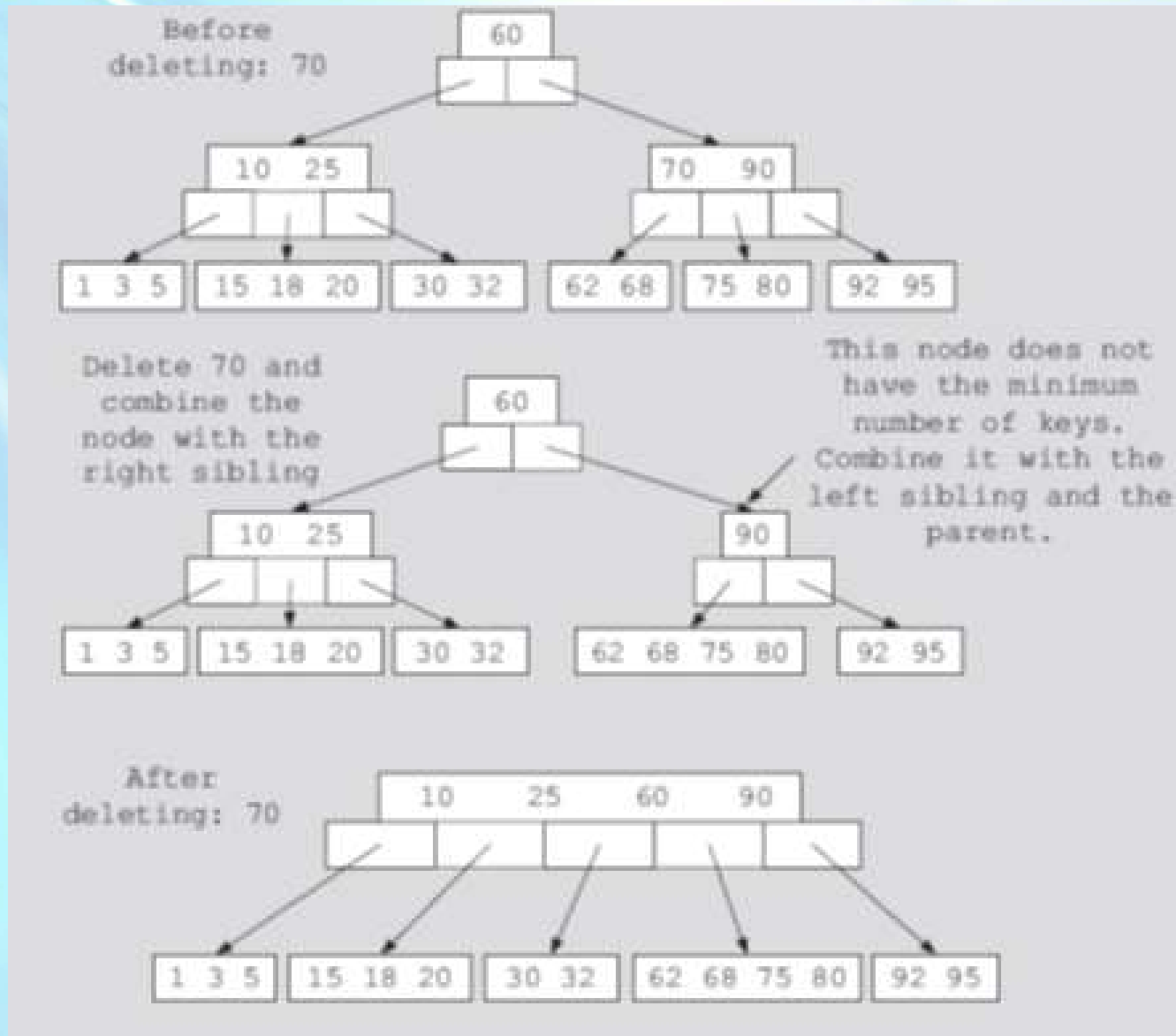


FIGURE 11-32 B-tree before and after deleting 30



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 khóa 30, 54, 70, 25