

Cấu trúc dữ liệu và giải thuật

CẤU TRÚC CÂY

Giảng viên:

Văn Chí Nam – Nguyễn Thị Hồng Nhung – Đặng Nguyễn Đức Tiến

Nội dung trình bày

2

Khái niệm

Phép duyệt cây và Biểu diễn cây

Cây nhị phân và Cây nhị phân tìm kiếm

Cây AVL

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

3

Khái niệm

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Một số thuật ngữ

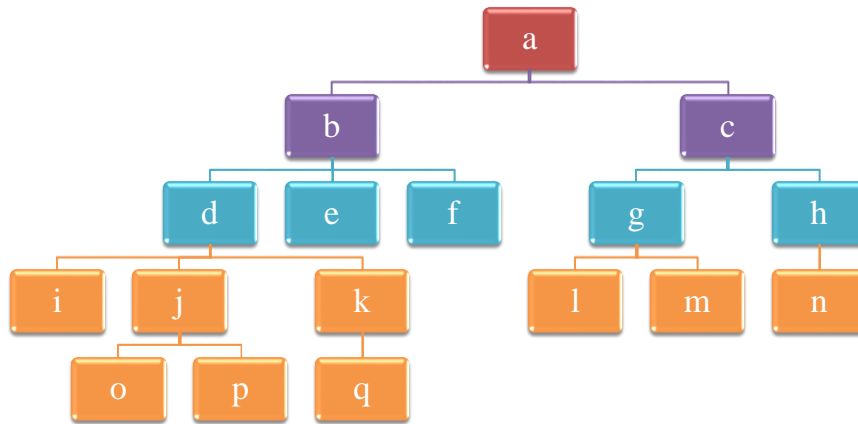
4

- Tree
- Search tree
- Binary search tree
- Balanced tree
- AVL tree
- AA tree
- Red-Black tree
- ...

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây tổng quát

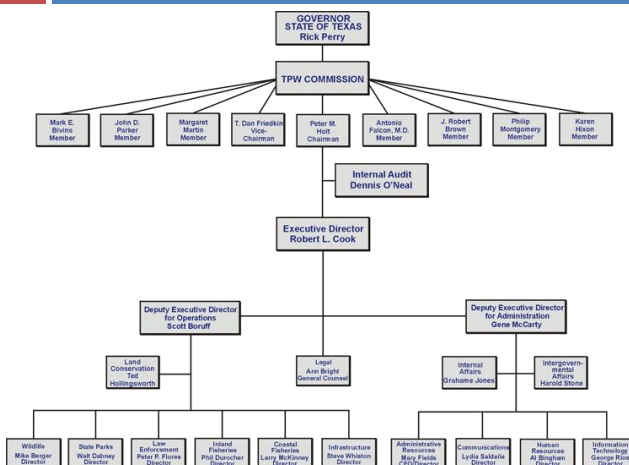
5



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

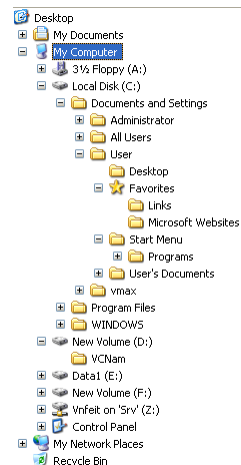
Cây tổng quát

6



Sơ đồ tổ chức

Cấu trúc dữ liệu và giải thuật - HCMUS 2015



Cây thư mục

Định nghĩa

7

Cây (cây có gốc) được xác định đệ quy như sau:

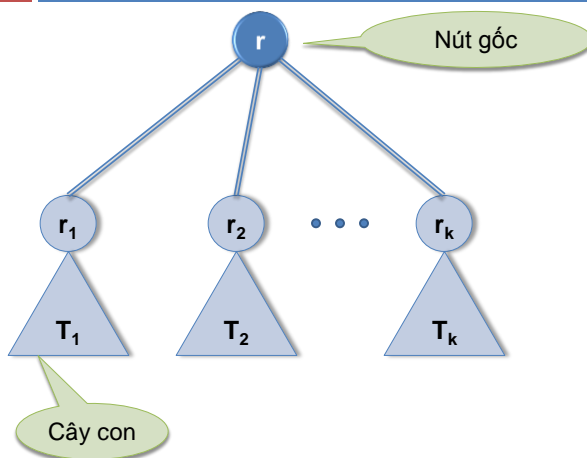
1. Tập hợp gồm 1 **đỉnh** là một cây. Cây này có **gốc** là đỉnh duy nhất của nó.
2. Gọi T_1, T_2, \dots, T_k ($k \geq 1$) là các cây không cắt nhau có gốc tương ứng r_1, r_2, \dots, r_k .

Giả sử r là một đỉnh mới không thuộc các cây T_i . Khi đó, tập hợp T gồm đỉnh r và các cây T_i tạo thành một cây mới với gốc r . Các cây T_1, T_2, \dots, T_k được gọi là cây con của gốc r .

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Định nghĩa

8



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các khái niệm

9

- node: đỉnh
- parent (của node n): node cha của node n .
Node **phía trên trực tiếp** của node n trong cây.
- child (của node n): node con của node n . Node **phía dưới trực tiếp** của node n trong cây.
- root: gốc cây. Node duy nhất không có node cha
- leaf: node lá. Node không có node con.
- path: đường đi

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các khái niệm

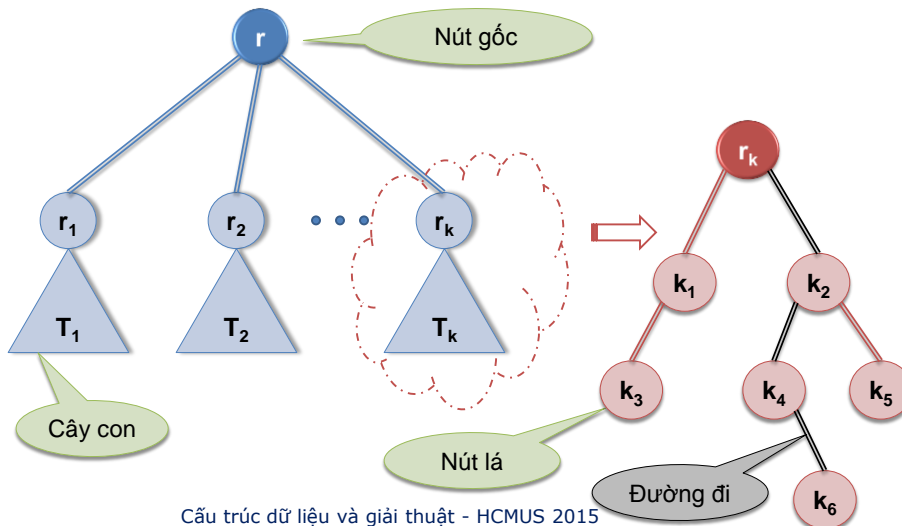
10

- siblings: các node cùng node cha.
- ancestor (của node n): node trên đường đi từ node gốc đến node n .
- descendant (của node n): node trên đường đi từ node n đến node lá.
- subtree (của node n): cây con. Cây bao gồm 1 node con của node n và các node “hậu duệ” của node này.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các khái niệm

11



Các khái niệm

12

- ◉ degree/order: bậc
 - ▣ Bậc của node: Số con của node
 - ▣ Bậc của cây: bậc lớn nhất trong số các node của cây.
- ◉ depth/level: độ sâu/mức
 - ▣ Mức (độ sâu) của node:
 - Nếu node n là node gốc:
 - ▣ $\text{level}(n) = 1$
 - Nếu node n không phải là node gốc:
 - ▣ $\text{level}(n) = 1 + \text{level}(\text{parent}(n))$.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các khái niệm

13

- ◉ **height: chiều cao.** Số lượng node trên đường đi **dài nhất** từ node gốc đến node lá.

- ▣ Chiều cao cây:

- Nếu cây T rỗng: $\text{height}(T) = 0$

- Nếu cây T khác rỗng: $\text{height}(T) = \max\{\text{level}(N_i)\}, N_i \in T$

- ▣ Chiều cao cây:

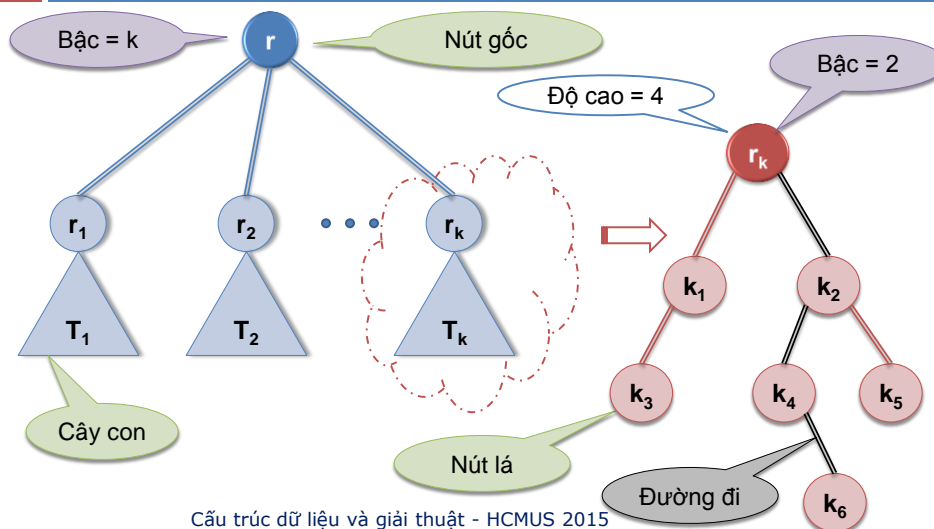
- Nếu cây T rỗng: $\text{height}(T) = 0$

- Nếu cây T khác rỗng: $\text{height}(T) = 1 + \max\{\text{height}(T_i)\}, T_i \text{ là cây con của } T$

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các khái niệm

14



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

- Đảm bảo đến mỗi node trên cây **chính xác một lần** một cách **có hệ thống**.
- Nhiều thao tác xử lý trên cây cần phải sử dụng đến phép duyệt cây.
- Các phép cơ bản:
 - ▣ Duyệt trước (Pre-order)
 - ▣ Duyệt giữa (In-order)
 - ▣ Duyệt sau (Post-order)

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

17

Tìm cha một đỉnh.

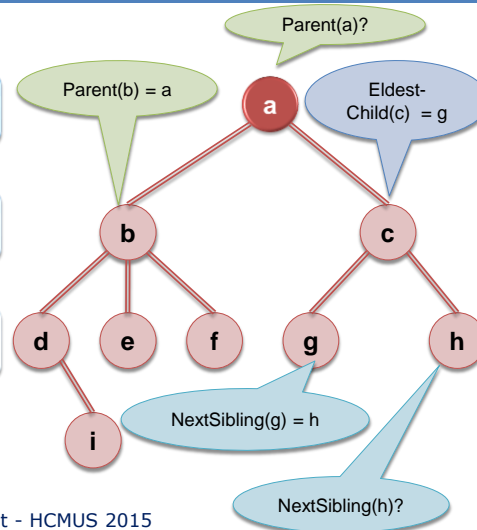
- $Parent(x)$

Tìm đỉnh con trái nhất.

- $EldestChild(x)$

Tìm đỉnh kề phải.

- $NextSibling(x)$



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

18

Duyệt trước

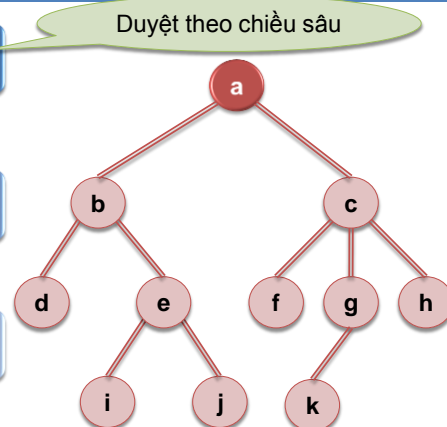
- $a b d e i j c f g k h$

Duyệt giữa

- $d b i e j a f c k g h$

Duyệt sau

- $d i j e b f k g h c a$



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

19

Pre-order

```
void Preorder(NODE A)
{
    NODE B;
    Visit(A);
    B = EldestChild(A);
    while (B != Ø) {
        Preorder(B);
        B = NextSibling(B);
    }
}
```

Post-order

```
void Postorder(NODE A)
{
    NODE B;

    B = EldestChild(A);
    while (B != Ø) {
        Postorder(B);
        B = NextSibling(B);
    }
    Visit(A);
}
```

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

20

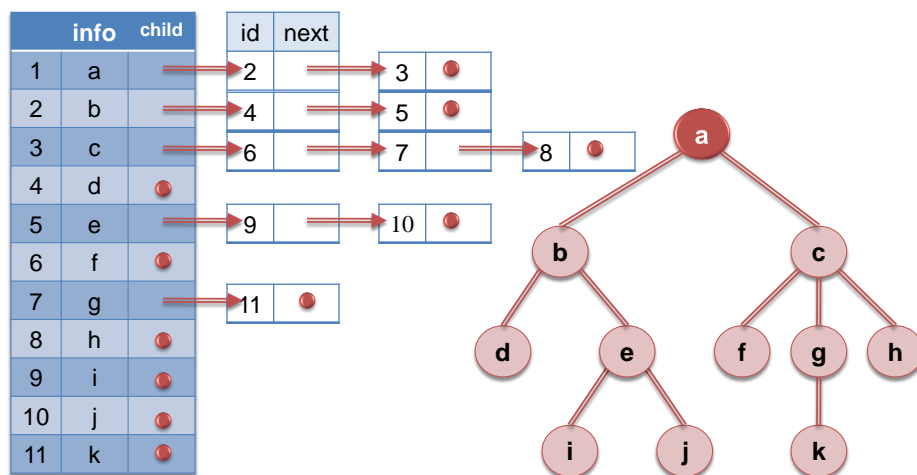
In-Order

```
void Inorder(NODE A)
{
    NODE B;
    B = EldestChild(A);
    if (B != Ø) {
        Inorder(B);
        B = NextSibling(B);
    }
    Visit(A);
    while (B != Ø) {
        Inorder(B);
        B = NextSibling(B);
    }
}
```

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

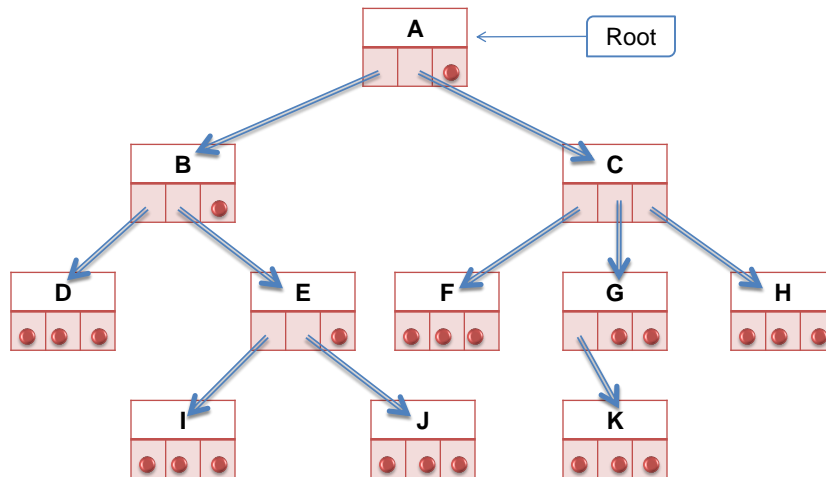
Biểu diễn cây

Bảng danh sách cây con



Bảng danh sách cây con

23

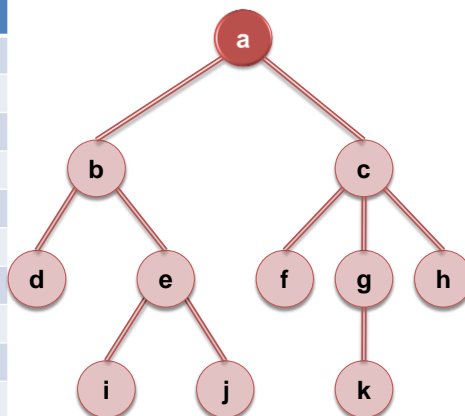


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Bảng định trái nhất và định kề phải

24

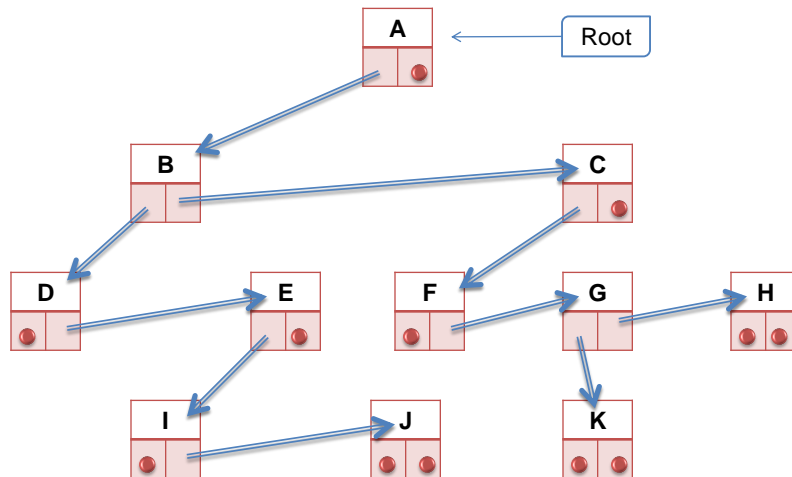
	Info	Eldest Child	Next Sibling
1	a	2	0
2	b	4	3
3	c	6	0
4	d	0	5
5	e	9	0
6	f	0	7
7	g	11	8
8	h	0	0
9	i	0	10
10	j	0	0
11	k	0	0



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Bảng đỉnh trái nhất và đỉnh kề phải

25

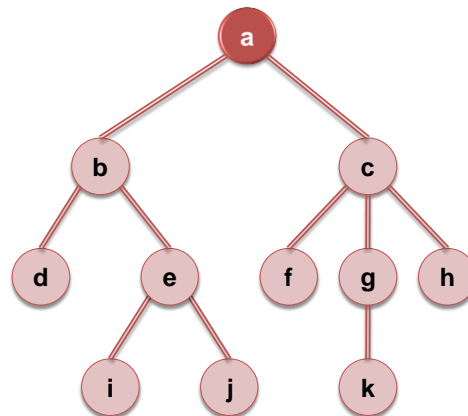


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Bảng cha mỗi đỉnh

26

	Info	Parent
1	a	0
2	b	1
3	c	1
4	d	2
5	e	2
6	f	3
7	g	3
8	h	3
9	i	5
10	j	5
11	k	7



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

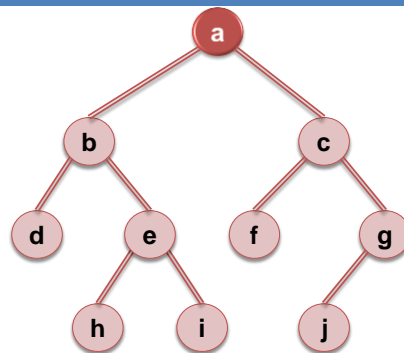
Cây nhị phân

Binary tree

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân

- Là cây mà mỗi đỉnh có bậc tối đa bằng 2.
- Các cây con được gọi là cây con trái và cây con phải.
- Có toàn bộ các thao tác cơ bản của cây.



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân

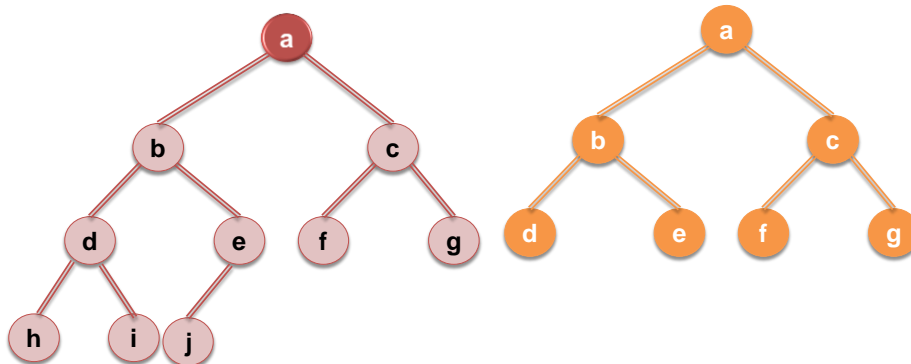
29

- **Cây nhị phân hoàn chỉnh (complete binary tree)**
 - ▣ Cây nhị phân có chiều cao là h thì có đầy đủ các node từ mức 1 đến mức $h-1$. Các node ở mức h sẽ được lấp từ trái sang phải.
- **Cây nhị phân đầy đủ (full binary tree)**
 - ▣ Cây nhị phân có chiều cao là h thì tất cả các node nằm ở mức từ 1 đến $h-1$ đều có 2 node con.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân

30



Cây nhị phân hoàn chỉnh

Cây nhị phân đầy đủ

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân

31

- ◉ Heap là một cây nhị phân hoàn chỉnh:
 - ▣ Max-heap: Node cha có giá trị **lớn hơn hoặc bằng** node con.
 - ▣ Min-heap: Node cha có giá trị **nhỏ hơn hoặc bằng** node con.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân

32

- ◉ **Chiều cao tối thiểu** của một cây nhị phân gồm N node?
- ◉ **Chiều cao tối đa** của một cây nhị phân gồm N node?

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân

33

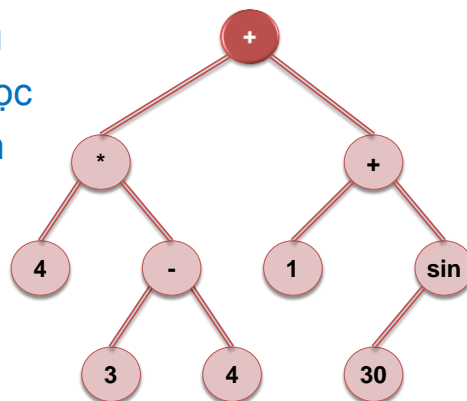
- **Số node tối thiểu** trên cây nhị phân có chiều cao h ?
- **Số node tối đa** trên cây nhị phân có chiều cao h ?

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Một số ứng dụng

34

- Cây tổ chức thi đấu
- Cây biểu thức số học
- Lưu trữ và tìm kiếm thông tin.



Cây biểu thức:
 $4 * (3 - 4) + (1 + \sin(30))$

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân tìm kiếm

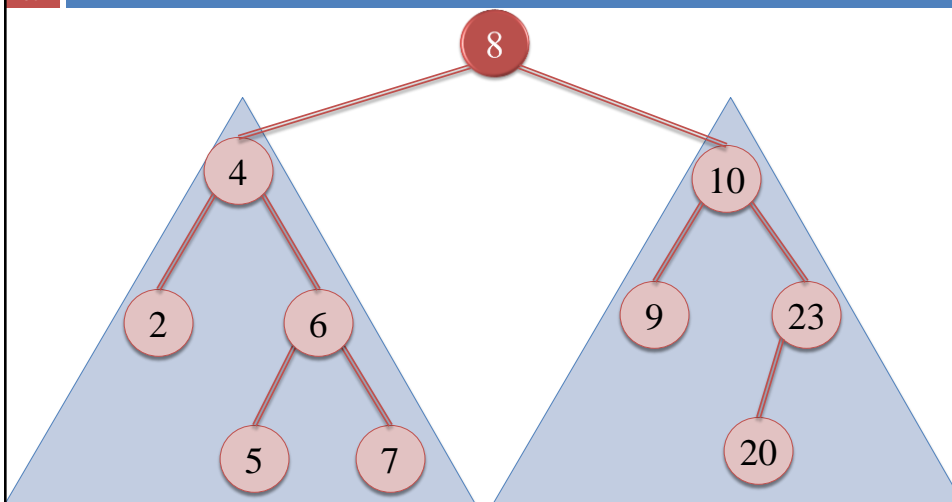
35

- Cây nhị phân tìm kiếm là cây nhị phân thỏa mãn các điều kiện sau:
 1. Khóa của node gốc **lớn hơn tất cả** khóa của các node thuộc cây con trái.
 2. Khóa của node gốc **nhỏ hơn tất cả** khóa của các node thuộc cây con phải.
 3. Cây con trái và cây con phải của node gốc là cây nhị phân tìm kiếm.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân tìm kiếm

36



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Cây nhị phân tìm kiếm

37

◉ Đặc điểm:

- ▣ Có thứ tự
- ▣ Không có phần tử trùng
- ▣ Dễ dàng tạo dữ liệu sắp xếp, và tìm kiếm

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

**Thao tác trên
cây nhị phân tìm kiếm**

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các thao tác

39

- Thêm phần tử (khóa)
- Tìm kiếm phần tử (khóa)
- Xóa phần tử (khóa)
- Sắp xếp
- Duyệt cây
- Quay cây

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Thêm phần tử

40

- Bước 1: Bắt đầu từ gốc
- Bước 2: So sánh dữ liệu (khóa) cần thêm với dữ liệu (khóa) của node hiện hành.
 - ▣ Nếu bằng nhau => Đã tồn tại. Kết thúc
 - ▣ Nếu nhỏ hơn => Đi qua nhánh trái, Tiếp bước 2.
 - ▣ Nếu lớn hơn => Đi qua nhánh phải, Tiếp bước 2.
- Bước 3: Không thể đi tiếp nữa => Tạo node mới với dữ liệu (khóa) cần thêm. Kết thúc

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Tìm kiếm phần tử

41

- Bước 1: Bắt đầu từ gốc
- Bước 2: So sánh dữ liệu (khóa) cần tìm với dữ liệu (khóa) của node hiện hành.
 - ▣ Nếu bằng nhau => Tìm thấy. Kết thúc
 - ▣ Nếu nhỏ hơn => Đi qua nhánh trái, Tiếp bước 2.
 - ▣ Nếu lớn hơn => Đi qua nhánh phải, Tiếp bước 2.
- Bước 3: Không thể đi tiếp nữa => Không tìm thấy. Kết thúc.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xóa phần tử

42

- Tìm đến node chứa dữ liệu (khóa) cần xóa.
- Xét các trường hợp:
 - ▣ Node lá
 - ▣ Node chỉ có 1 con
 - ▣ Node có 2 con: dùng phần tử thế mạng để xóa thế.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Sắp xếp

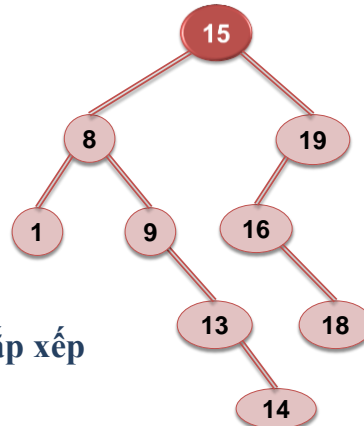
43

- Cho cây nhị phân tìm kiếm

- Thứ tự duyệt các node nếu sử dụng Duyệt giữa?

- Nêu nhận xét

- ▣ Có thể dễ dàng tạo dữ liệu sắp xếp nếu dùng phép duyệt giữa

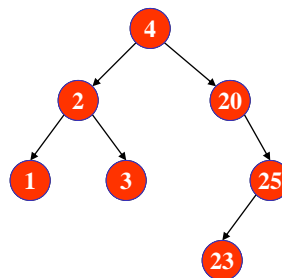


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

44

- Duyệt trước

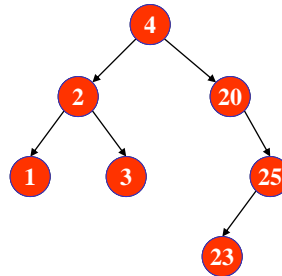


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

45

◉ Duyệt giữa

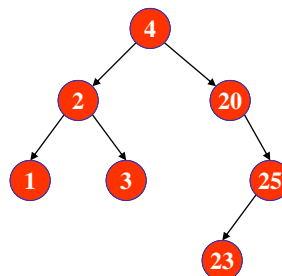


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép duyệt cây

46

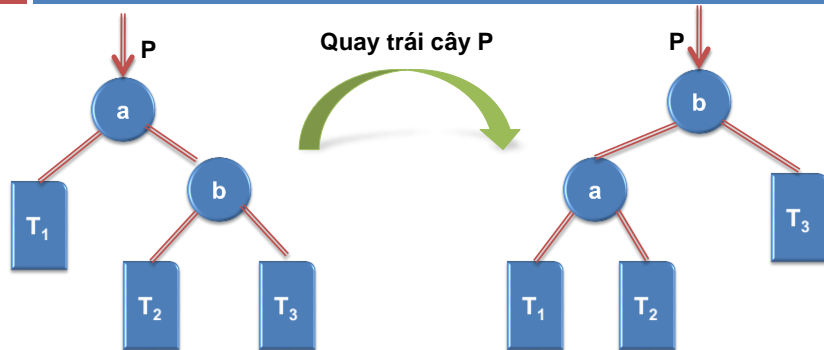
◉ Duyệt sau



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép quay trái

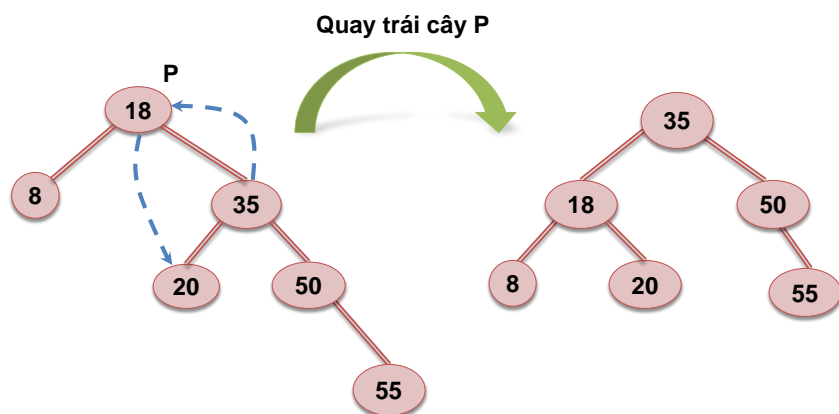
47



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép quay trái

48



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép quay phải

49

Quay phải cây P

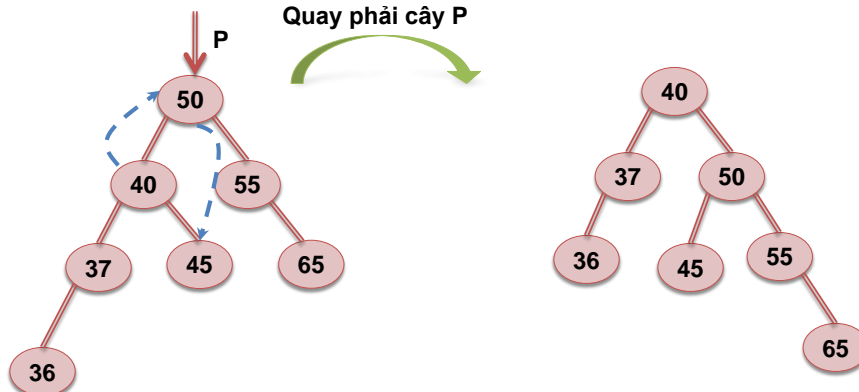


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Phép quay phải

50

Quay phải cây P



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Thời gian thực hiện các phép toán

51

- ◉ Đối với phép tìm kiếm:
 - ▣ Trường hợp tốt nhất: mỗi nút (trừ nút lá) đều có 2 con: $O(\log_2 n)$ (chính là chiều cao của cây).
 - ▣ Trường hợp xấu nhất: cây trở thành danh sách liên kết: $O(n)$.
 - ▣ Trường hợp trung bình là bao nhiêu?
 $O(\log_2 n)$

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Ví dụ

52

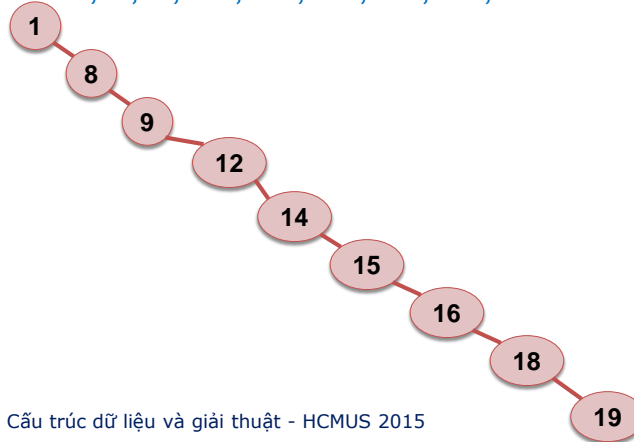
- ◉ Tạo cây nhị phân tìm kiếm theo thứ tự nhập như sau: 1, 8, 9, 12, 14, 15, 16, 18, 19

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Ví dụ

53

- ◉ Tạo cây nhị phân tìm kiếm theo thứ tự nhập như sau: 1, 8, 9, 12, 14, 15, 16, 18, 19



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

54

Cây AVL

AVL tree

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Giới thiệu

55

- Do G.M. Adelsen Velskii và E.M. Lendis đưa ra vào năm 1962, đặt tên là cây AVL.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Định nghĩa

56

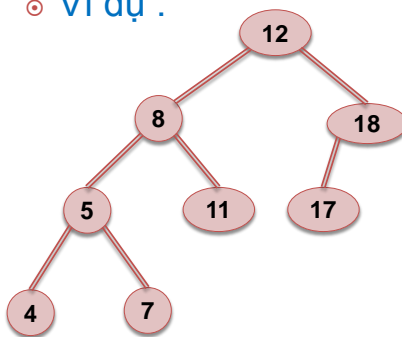
- Cây cân bằng AVL là cây nhị phân tìm kiếm mà tại mỗi đỉnh của cây, độ cao của cây con trái và cây con phải **không chênh lệch quá 1**.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

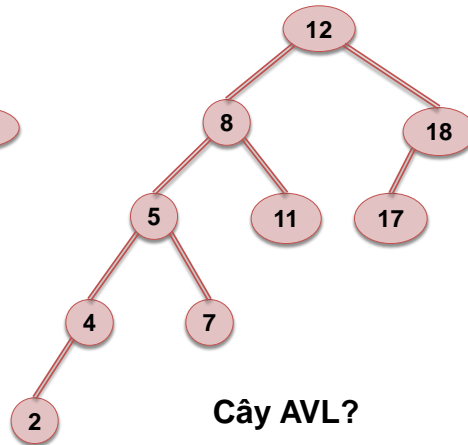
Cây AVL

57

◦ Ví dụ :



Cây AVL?



Cây AVL?

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xây dựng cây cân bằng

58

- Việc xây dựng cây cân bằng dựa trên cây nhị phân tìm kiếm, chỉ bổ sung thêm 1 giá trị cho biết sự cân bằng của các cây con như thế nào.
- Cách làm gợi ý:

```
struct NODE {  
    Data key;  
    NODE *pLeft, *pRight;  
    int bal;  
};
```
- Trong đó giá trị bal (balance, cân bằng) có thể là: 0: cân bằng; 1: lệch trái; 2: lệch phải

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các trường hợp mất cân bằng

59

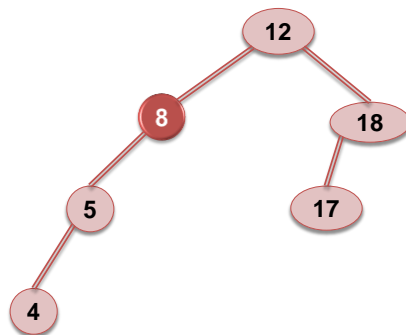
- Mất cân bằng trái-trái (L-L)
- Mất cân bằng trái-phải (L-R)
- Mất cân bằng phải-phải (R-R)
- Mất cân bằng phải-trái (R-L)

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các trường hợp mất cân bằng

60

- Mất cân bằng trái-trái (L-L)

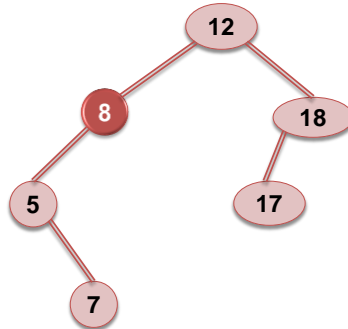


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các trường hợp mất cân bằng

61

◉ Mất cân bằng trái-phải (L-R)

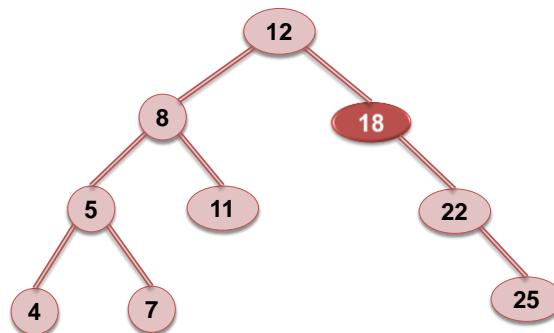


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các trường hợp mất cân bằng

62

◉ Mất cân bằng phải-phải (R-R)

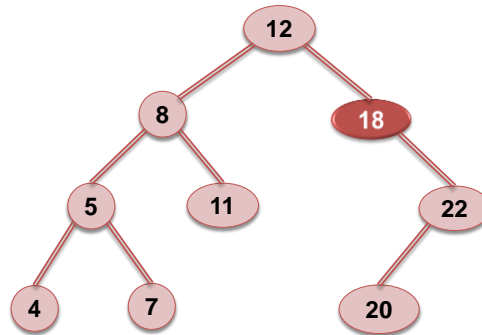


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Các trường hợp mất cân bằng

63

◦ Mất cân bằng phải-trái (R-L)



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

64

◦ Giả sử tại một node cây xảy ra mất cân bằng bên phải (cây con phải chênh lệch với cây con trái hơn một đơn vị):

▣ Mất cân bằng phải-phải (RR)

- Quay trái

▣ Mất cân bằng phải-trái (R-L)

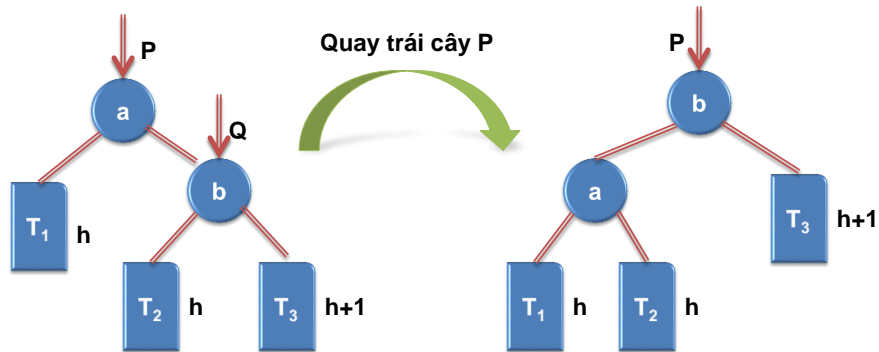
- Quay phải
- Quay trái

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

65

◉ P mất cân bằng phải-phải (RR):

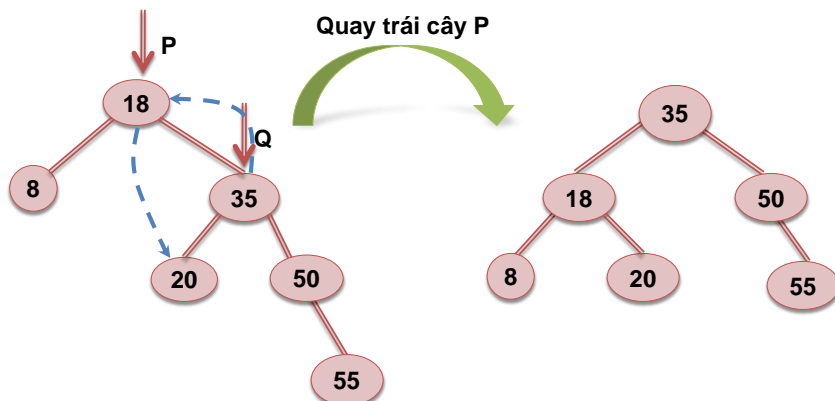


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

66

◉ P mất cân bằng phải-phải (RR):



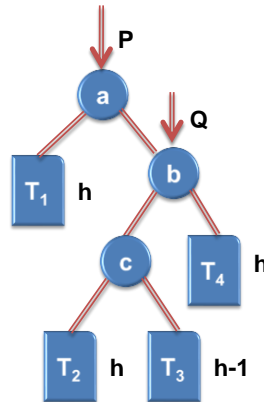
Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

67

◉ P mất cân bằng phải-trái (RL):

- ▣ Bước 1: quay phải Q
- ▣ Bước 2: quay trái cây P



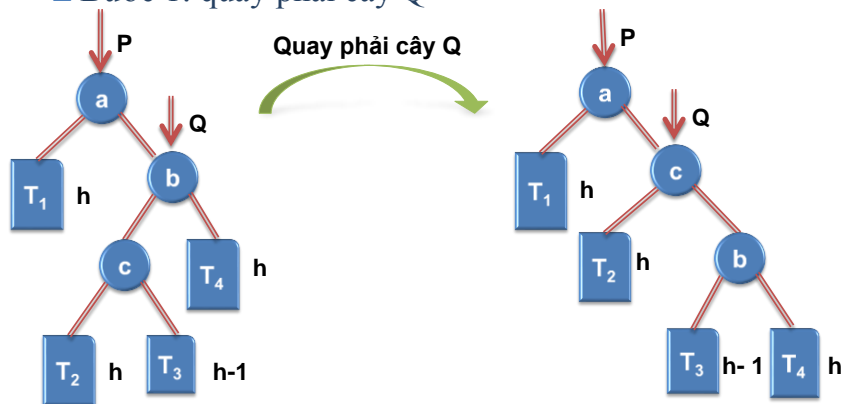
Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

68

◉ P mất cân bằng phải-trái (RL):

- ▣ Bước 1: quay phải cây Q



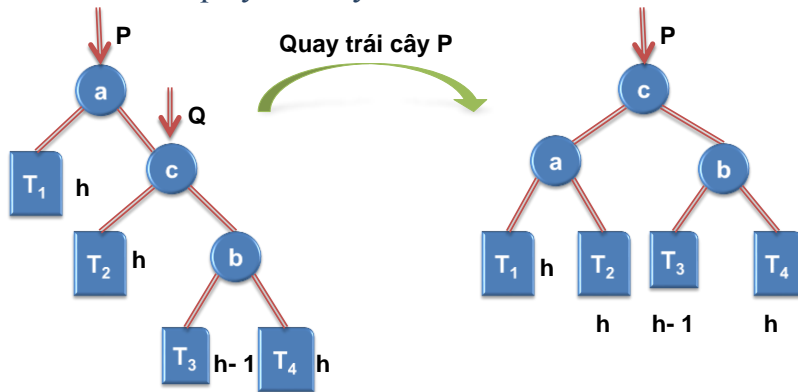
Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

69

◉ P mất cân bằng phải-trái (RL):

▣ Bước 2: quay trái cây P

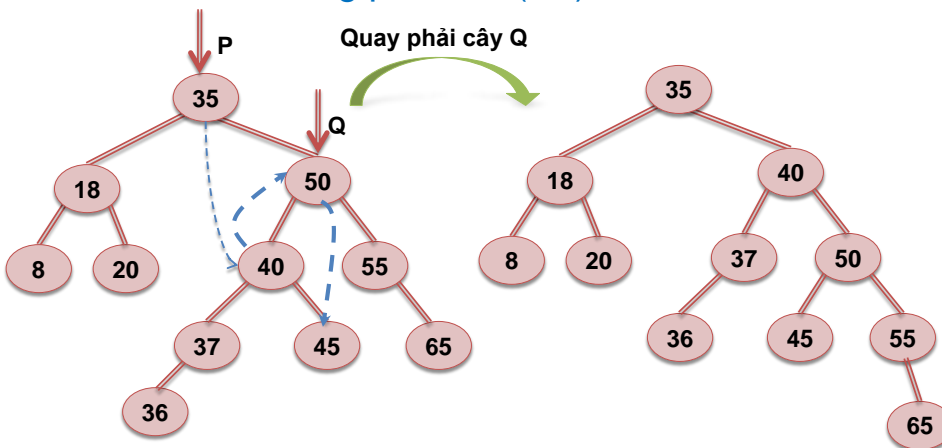


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

70

◉ P mất cân bằng phải-trái (RL) – Bước 1:

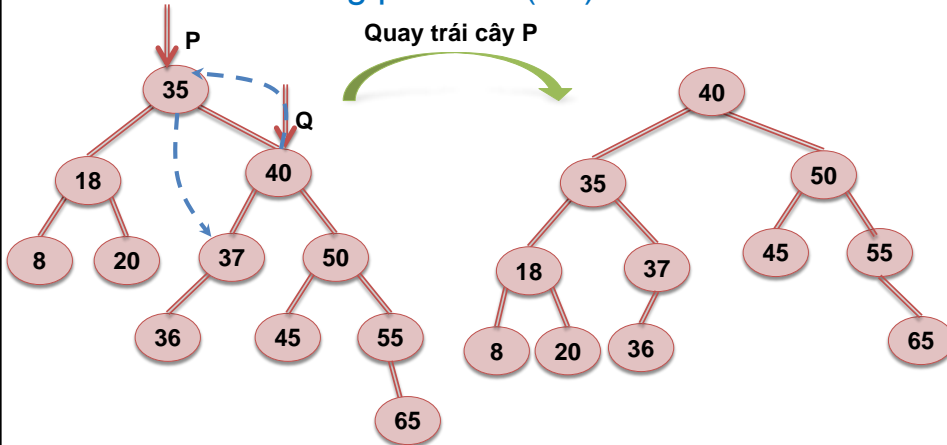


Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

71

◉ P mất cân bằng phải-trái (RL) - Bước 2:



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

72

◉ Khi một node cây xảy ra mất cân bằng bên trái (cây con trái chênh lệch với cây con phải hơn một đơn vị): (thực hiện đối xứng với trường hợp mất cân bằng bên phải)

▣ Mất cân bằng trái-trái (LL)

- Quay phải

▣ Mất cân bằng trái-trái (L-R)

- Quay trái
- Quay phải

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Xử lý mất cân bằng

73

There are 4 cases in all, choosing which one to rotate by seeing the direction of the first 2 nodes from the unbalanced node to the newly inserted node and matching them to the top most row.

Root is the initial parent before a rotation and Pivot is the child to take the root's place.



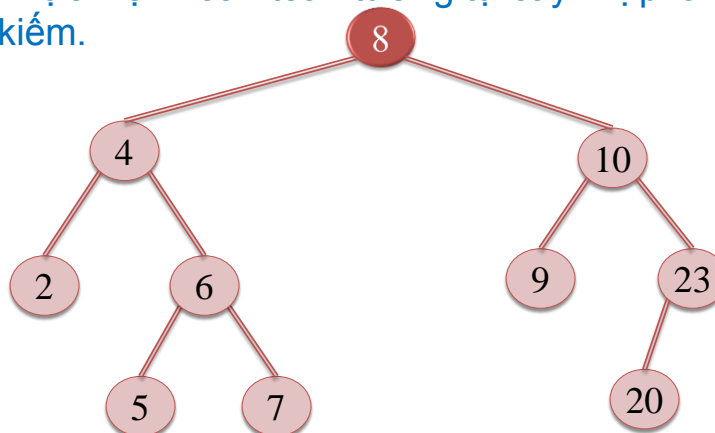
Theo Wikipedia

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Thao tác tìm kiếm

74

- Thực hiện hoàn toàn tương tự cây nhị phân tìm kiếm.



Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Thao tác thêm phần tử

75

- Thực hiện tương tự với việc thêm phần tử của cây nhị phân tìm kiếm.
- Nếu xảy ra việc mất cân bằng thì xử lý bằng các trường hợp mất cân bằng đã biết.

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Thao tác xóa phần tử

76

- Thực hiện tương tự cây nhị phân tìm kiếm: xét 3 trường hợp, và tìm phần tử thế mạng nếu cần.
- Sau khi xóa, nếu cây mất cân bằng, thực hiện cân bằng cây.
- Lưu ý: *việc cân bằng sau khi hủy có thể xảy ra dây chuyền.*

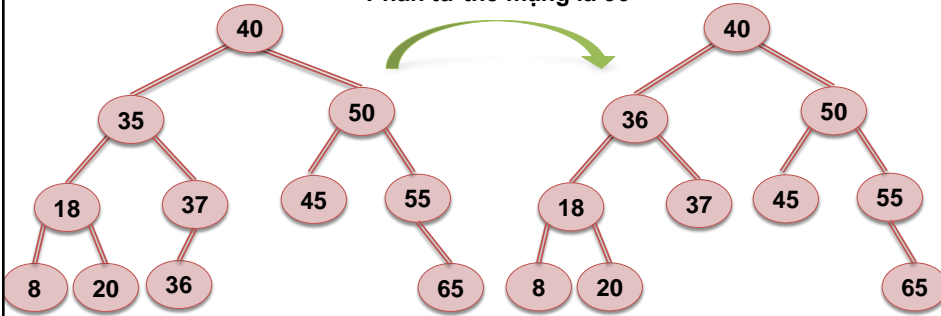
Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Thao tác xóa phần tử

77

◉ Ví dụ: xóa 35

Phần tử thế mạng là 36



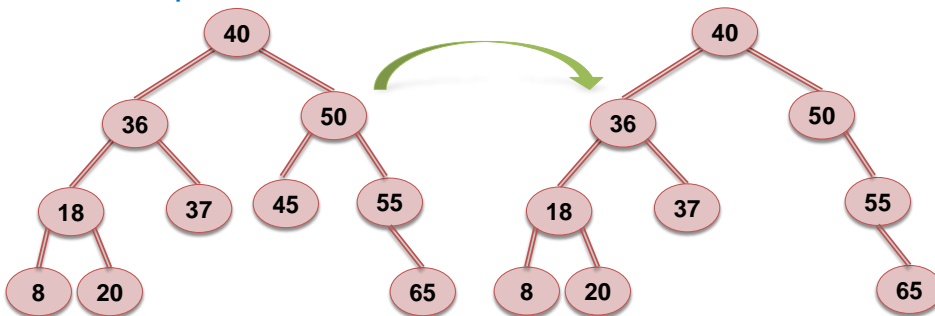
Cây vẫn cân bằng nên
không phải hiệu chỉnh

Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Thao tác xóa phần tử

78

◉ Xóa phần tử 45



Node 50 bị lệch phải !!!

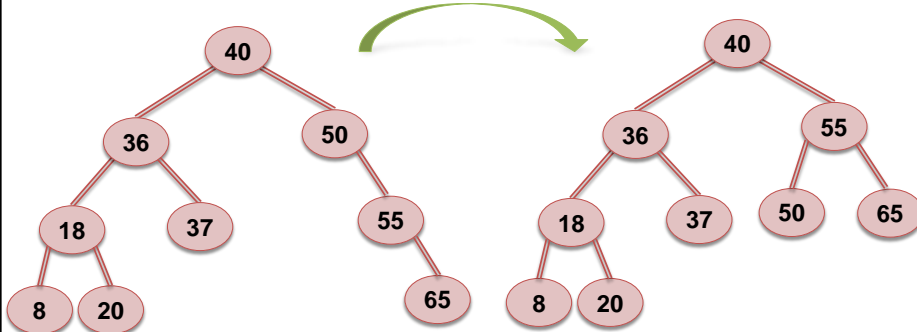
Cấu trúc dữ liệu và giải thuật - HCMUS 2015

Thao tác xóa phần tử

79

- Xóa phần tử 45: cân bằng lại cây

Quay trái tại node 50



Cấu trúc dữ liệu và giải thuật - HCMUS 2015