

**CẤU TRÚC DỮ LIỆU & GIẢI THUẬT**  
**HOMEWORK 5.3 – Cây nhị phân tìm kiếm, Priority Queue**  
 ---oOo---

**1. Cây nhị phân là:**

- Cây có tối đa 2 node
- Cây mà node gốc có tối đa 2 cây con
- Cây mà mỗi node phải có 2 cây con khác rỗng
- Cây mà mỗi node có 2 cây con

**2. Cho một cấu trúc cây:**

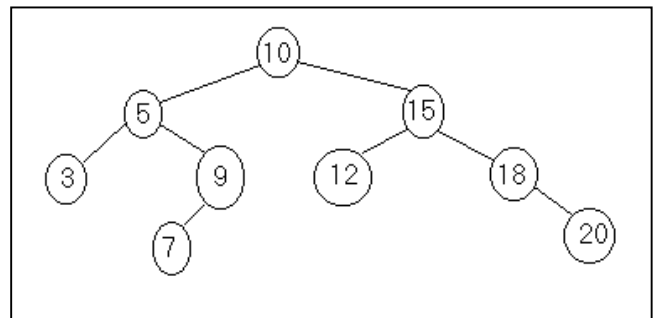
- Bậc (degree) của node gốc luôn là 0  
☐ Đúng ☐ Sai
- Các node không phải là gốc cũng không phải là lá thì gọi là các node trong (internal node)  
☐ Đúng ☐ Sai

**3. Cho một cây nhị phân như hình vẽ bên.**

Giả sử kết quả của phép duyệt cây là:

3 7 9 5 12 20 18 15 10

Hãy cho biết tên của phép duyệt cây?



**4. Vẽ cây nhị phân tìm kiếm sau khi chèn các khóa sau vào cây theo thứ tự từ trái qua phải. Cây ban đầu là rỗng.**

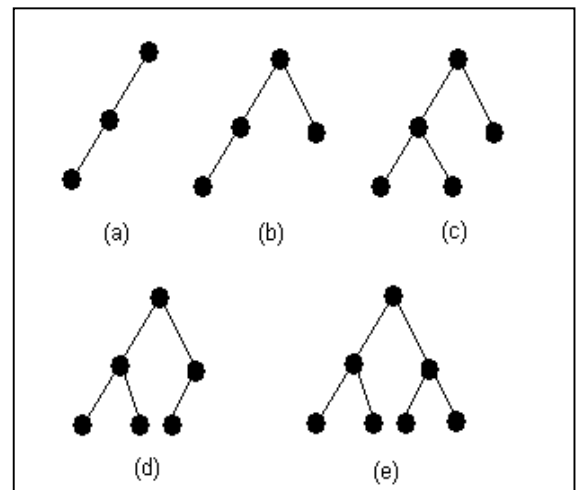
Key = {17, 9, 26, 12, 11, 7, 30, 20, 21, 10}

Sử dụng kết quả trên, vẽ lại cây BST khi xóa khóa 17.

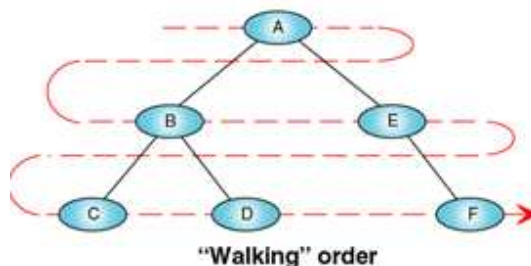
**5. Trong hình bên cạnh:**

- Cây nào là cây complete?
- Cây nào là cây full?

**6. Chứng minh rằng một cây nhị phân với chiều cao  $h$  sẽ có tối đa  $2^h - 1$  node**



**7. Viết giải thuật (mã giả) duyệt cây theo mức**

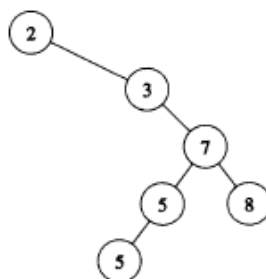


**8. Hãy điền các thuật ngữ tiếng Anh vào chỗ trống:**

- Cây nhị phân: .....
- Cây nhị phân đầy đủ: .....
- Cây nhị phân hoàn chỉnh: .....
- Cây con: .....
- Node lá: .....
- Node gốc: .....
- Node cha: .....
- Node anh em: .....
- Node nội (node trong): .....

**9. Cây nhị phân có chiều cao  $h=6$  thì sẽ có tối đa ..... node**

**10. Đây có phải là cây BST không?**



**11. Hãy vẽ hình cây BST khi thực hiện thêm lần lượt các khóa theo thứ tự:**

leopard, cobra, shark, horse, alligator, bat, tiger, cow, cat, dog, chicken, bear

**12. Hãy viết một hàm tạo DSLK đơn từ cây BST cho trước, sao cho giá trị các phần tử trong DSLK có thứ tự giảm dần. Trình bày ý tưởng và viết thành hàm bằng C/C++.**

**13. Cho một cây BST, mỗi node của cây là một số nguyên.**

Nếu áp dụng cách duyệt cây NLR ta có kết quả sau:

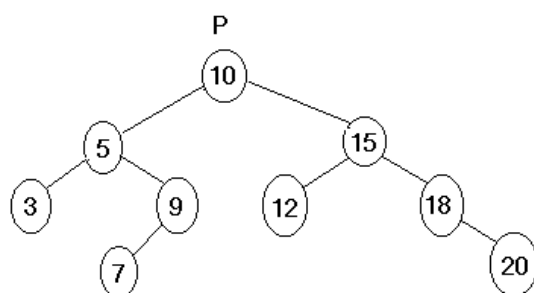
5 6 3 7 9 2 0 8 4

Nếu áp dụng cách duyệt cây LNR ta có kết quả sau:

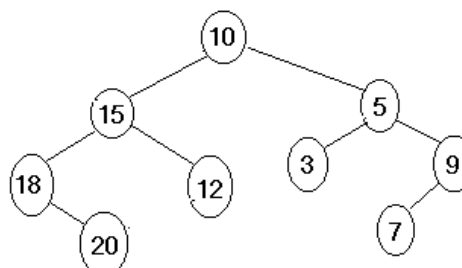
7 2 4 3 5 9 6 1 0

Hãy vẽ cây trên.

14. Cho một cây nhị phân như hình 1. Cho biết trước node P.



Hình 1



Hình 2

Hãy viết các lệnh cần thiết để chuyển cây sang dạng 2.

15. Cho một cây nhị phân có chứa các node sau (mỗi khóa xuất hiện 1 lần):

8 5 4 0 1 3 7 9 2

Áp dụng cách duyệt cây NLR để tính tổng các node lá, ta có kết quả tổng = 22

Áp dụng cách duyệt cây LNR để tính tổng các node không phải là node lá, ta có kết quả tổng = 18

Hãy chỉ ra các node lá của cây.

16. Hãy viết hàm “Tìm phần tử lớn nhất trong cây BST”:

- Dùng đệ qui
- Không dùng đệ qui

17. Vẽ cây BST khi thêm các key sau theo thứ tự vào 1 cây ban đầu rỗng

E A S Y Q U E S T I O N

18. Cây BST luôn luôn có chi phí tìm kiếm là  $O(\log_2 N)$

- ☐ Đúng ☐ Sai

19. Cho biết kết quả duyệt cây ở slide #77, chương 5 theo mỗi cách: preorder, inorder, postorder?

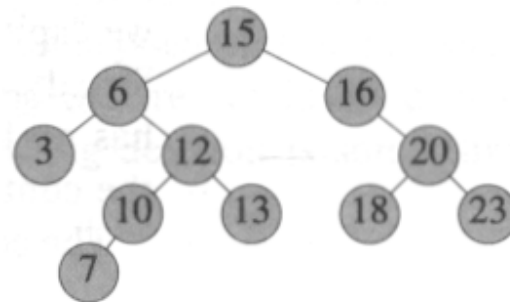
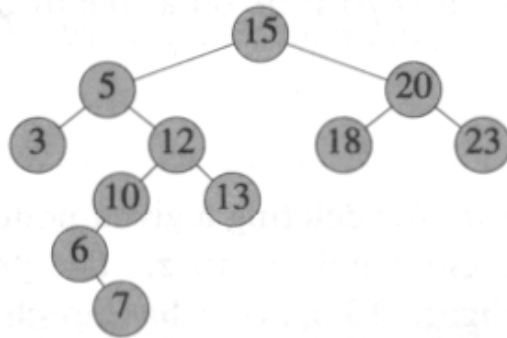
20. Căn cứ vào class BINARY\_TREE ở slide #78, chương 5, hãy cài đặt chi tiết các hàm sau:

- countNode: đếm số node trong cây
- height: tính chiều cao của cây

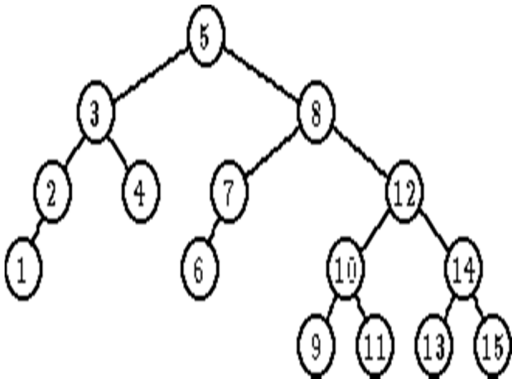
21. Căn cứ vào class BINARY\_SEARCH\_TREE ở slide #94, chương 5, hãy bổ sung và cài đặt chi tiết các hàm sau:

- BSTNode<T> \*maxNode(): trả về node có khóa lớn nhất, dùng đệ qui
- BSTNode<T> \*minNode(): trả về node có khóa nhỏ nhất, không dùng đệ qui

22. Hình 1: tìm các phần tử  $p_{tt}$  cho node “5”. Hình 2: tìm các phần tử  $p_{tt}$  cho node “15”?

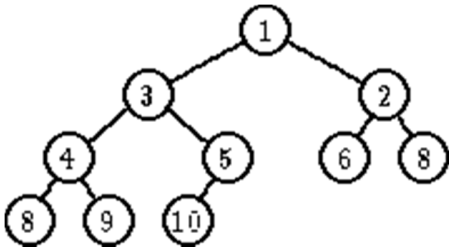


23. Cho cây BST như hình 1. Hãy vẽ cây sau khi thực hiện xóa node 8. Áp dụng 2 cách tìm phần tử thay thế.

| Hình 1  | Sau khi xóa 8, dùng $p_{tt}$ cách 1 | Sau khi xóa 8, dùng $p_{tt}$ cách 2 |
|---|-------------------------------------|-------------------------------------|
|  |                                     |                                     |

24. Cho priority queue như hình 1. Hãy cho biết kết quả của các thao tác sau (mỗi thao tác độc lập nhau, cùng thực hiện trên hình 1)

- `insert(0)`
- `deleteMin()`

| Hình 1   | Sau khi <code>insert(0)</code> | Sau khi <code>deleteMin()</code> |
|--|--------------------------------|----------------------------------|
|  |                                |                                  |

**25. Minh họa bằng hình vẽ khi thực hiện thao tác insert lần lượt các khóa vào Priority Queue:**  
15, 13, 9, 5, 12, 8, 7, 4, 0, 6, 2, 1

**26. Minh họa bằng hình vẽ khi thực hiện thao tác deleteMin ba lần liên tiếp trên Priority Queue của bài tập 25.**

--- Hết ---