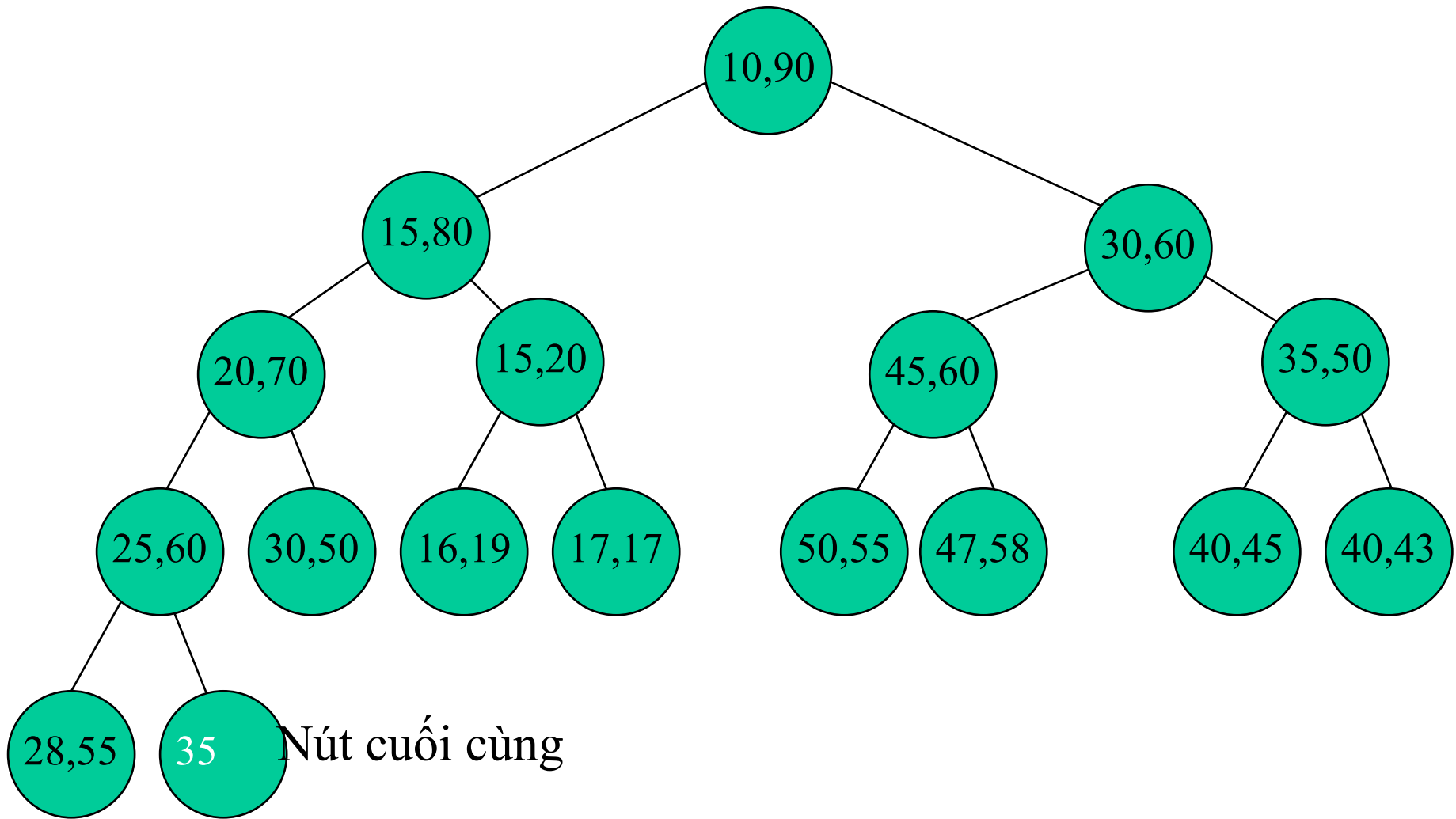


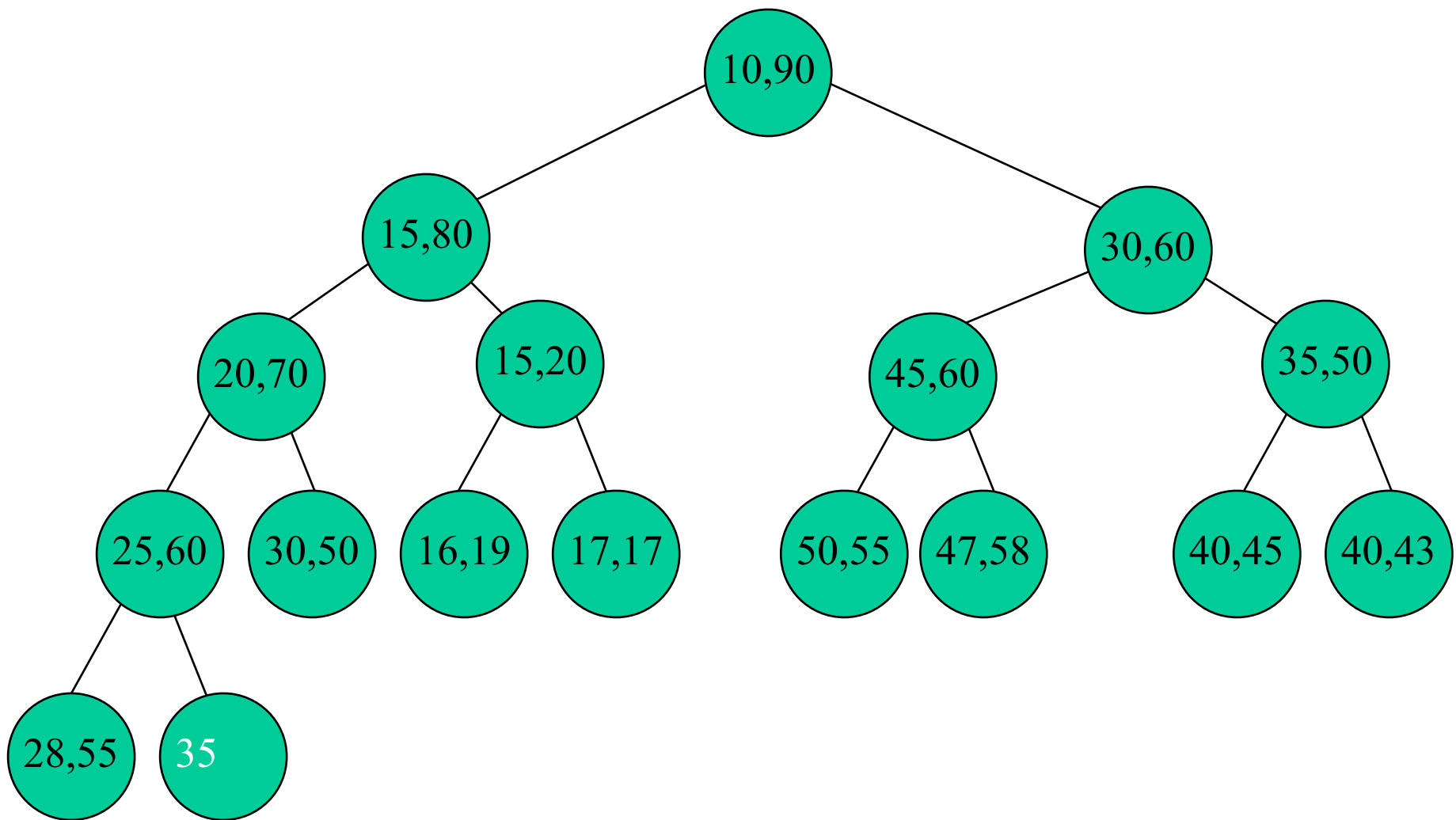
Lecture 4: Interval Heaps

- Là cây nhị phân gần hoàn chỉnh (*almost complete binary tree* - cây nhị phân mà mỗi đỉnh của nó phát sinh nút con trái trước, nút con phải sau).
- Mỗi nút (trừ nút cuối cùng) có 2 phần tử.
- Nút cuối cùng có 1 hoặc 2 phần tử.
- Gọi a và b là hai phần tử trong nút P , thì $a \leq b$.
- $[a, b]$ là đoạn được thể hiện trong P .
- Đoạn được thể hiện trong một nút chỉ có một phần tử a là đoạn $[a, a]$.
- Ta nói đoạn $[c, d]$ thuộc đoạn $[a, b]$ iff $a \leq c \leq d \leq b$.
- Trong **interval heap** đoạn của mỗi nút (trừ nút gốc) thuộc đoạn của nút cha của nó.



Giá trị cận bên trái định nghĩa một min heap.

Giá trị cận bên phải định nghĩa một max heap.

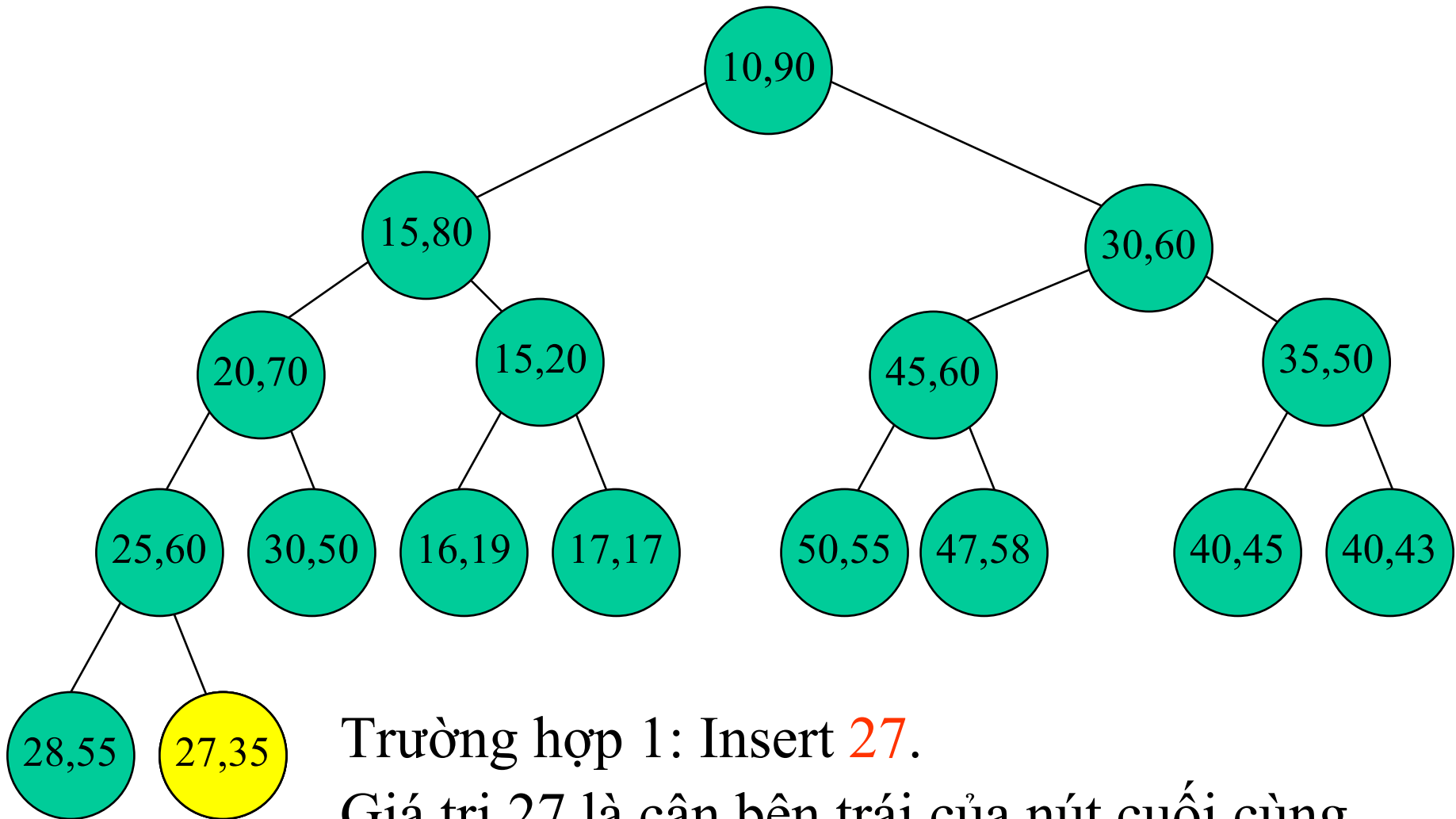


Giá trị Min và giá trị max nằm ở nút gốc

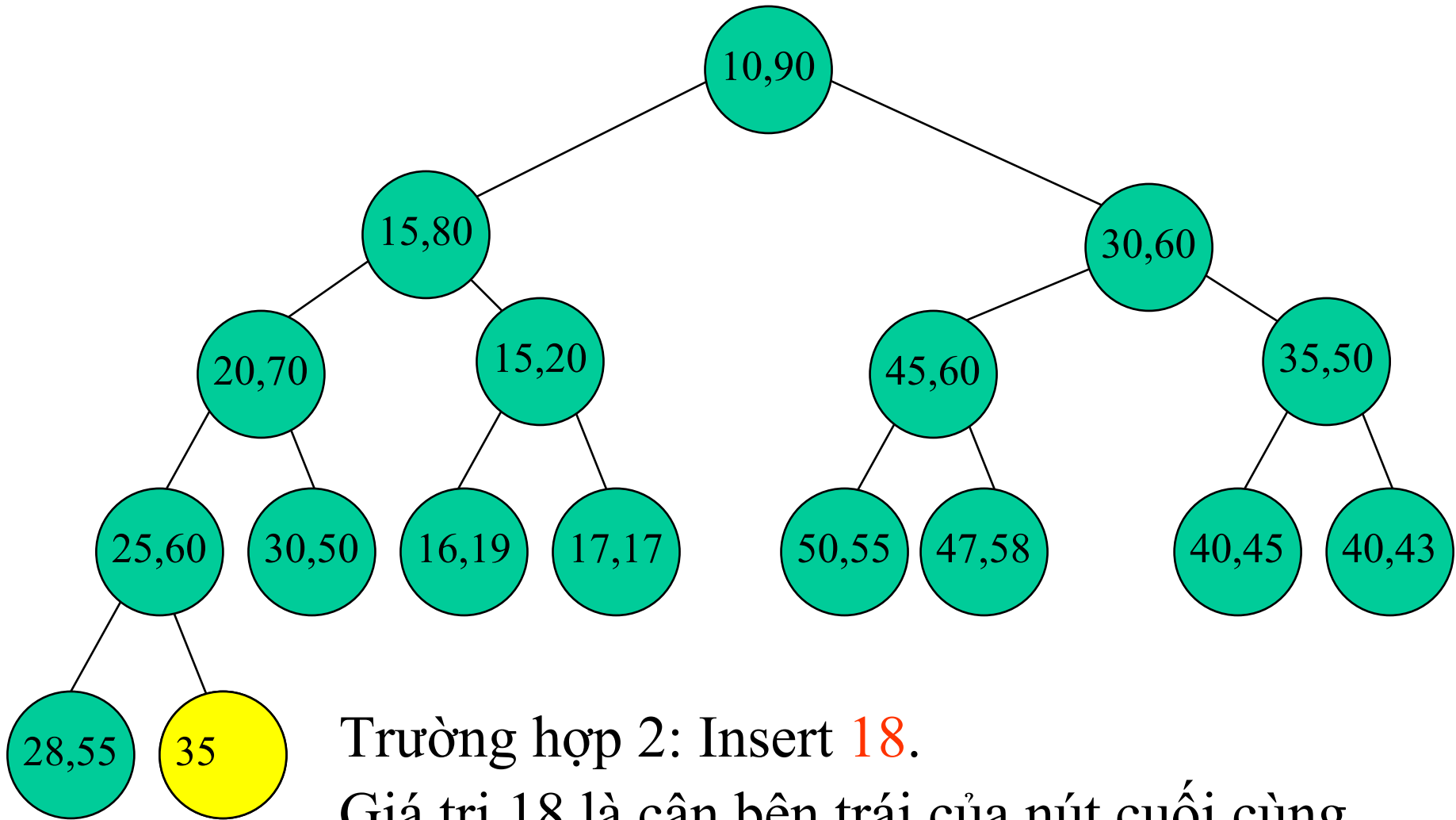
Store as an array.

Height is $\sim \log_2 n$.

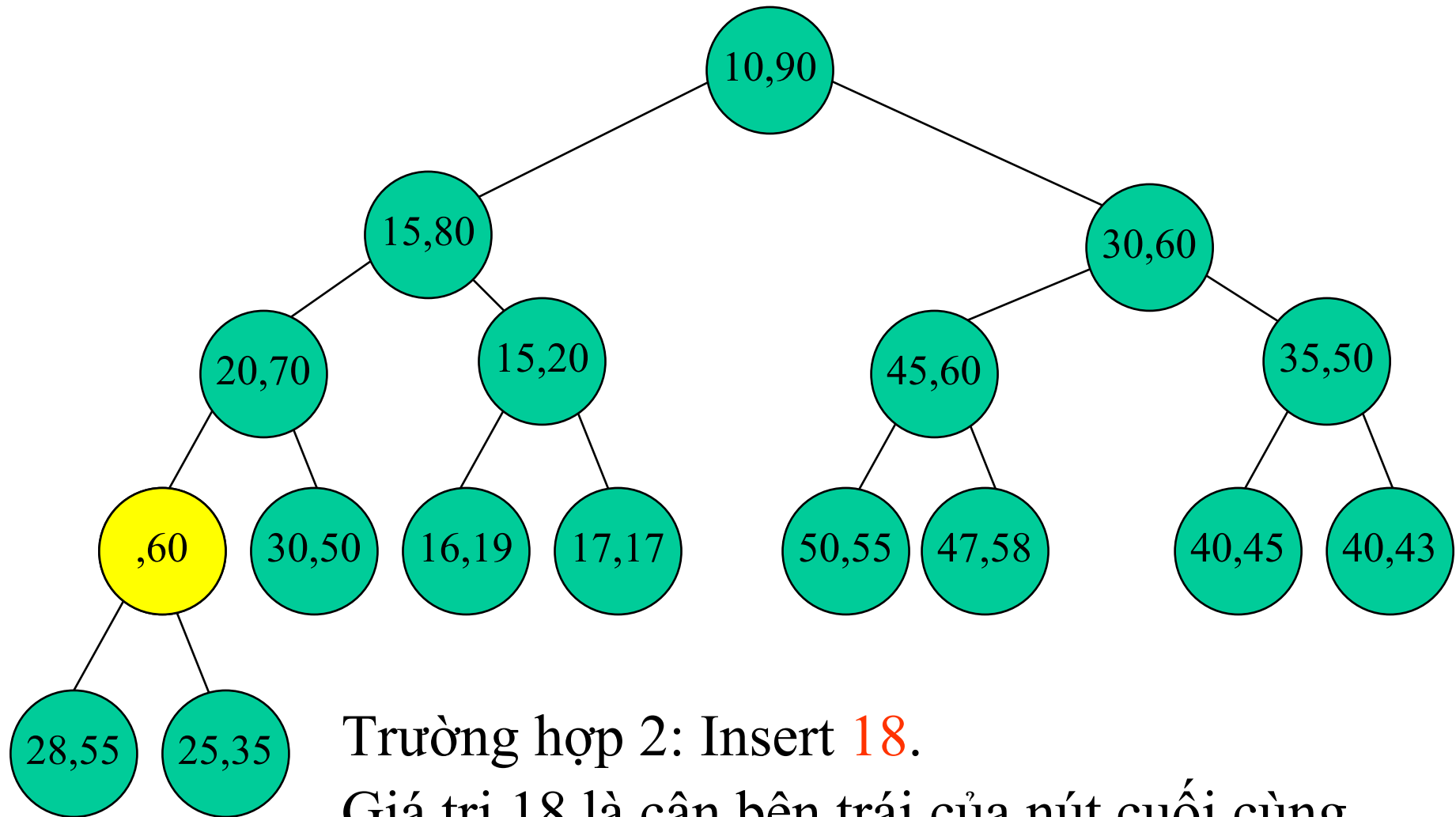
Insert An Element



Another Insert



Giá trị 18 phải đẩy dần lên trên để đảm bảo tính chất của min heap

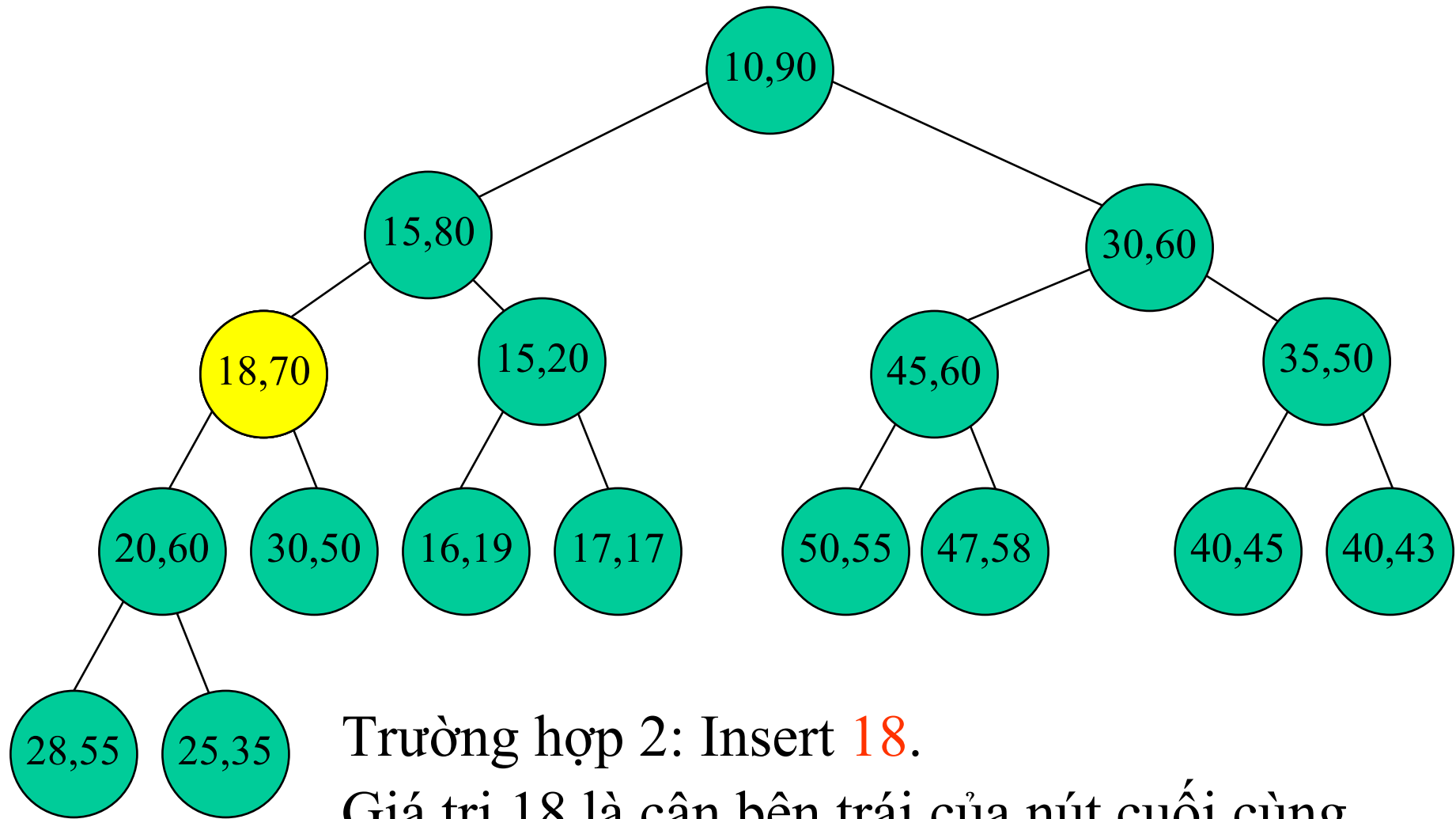


Trường hợp 2: Insert 18.

Giá trị 18 là cận bên trái của nút cuối cùng

Giá trị 18 này cũng được chèn vào min heap.

Giá trị 18 phải đẩy dần lên trên để đảm bảo tính chất của min heap

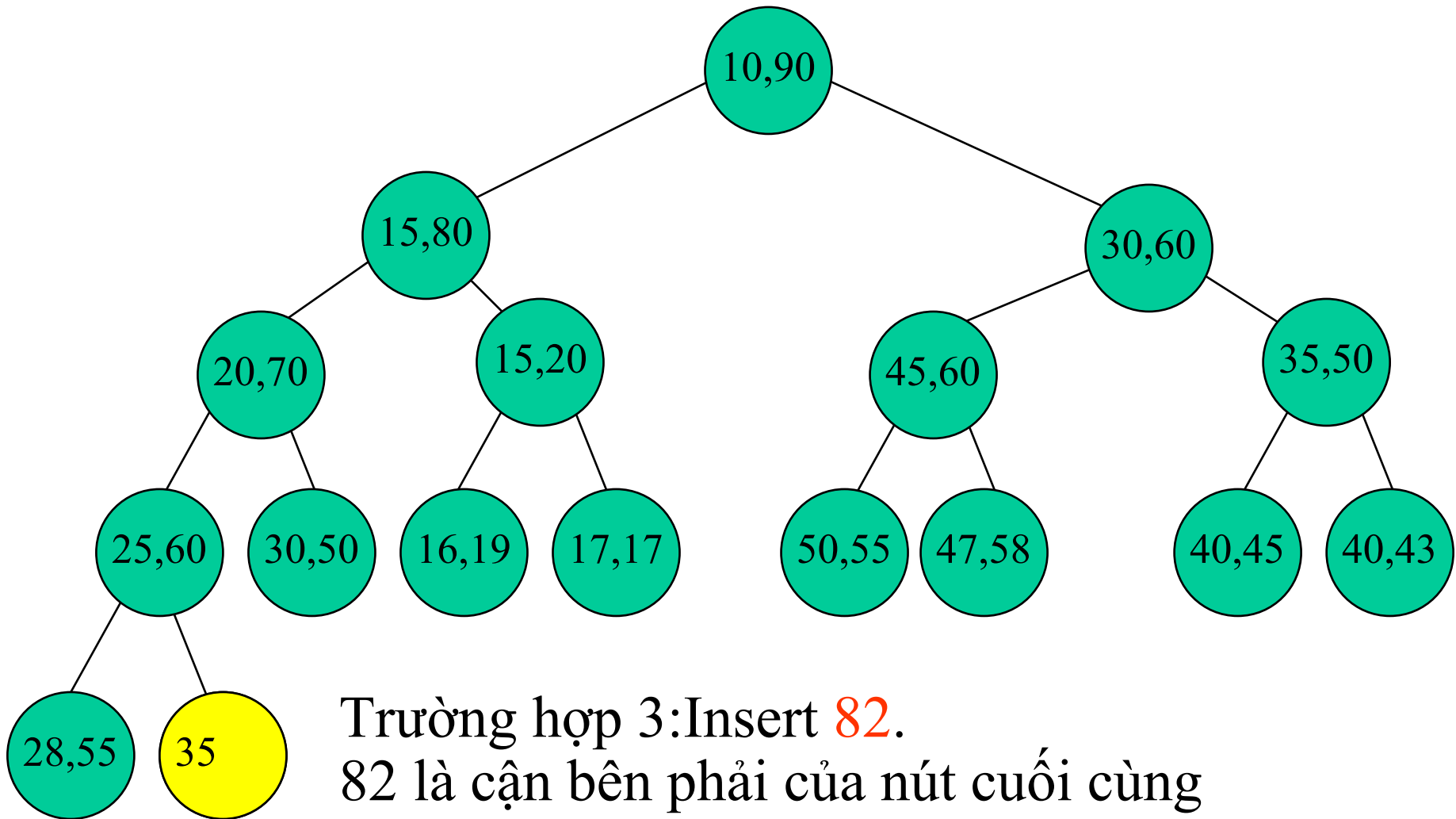


Trường hợp 2: Insert 18.

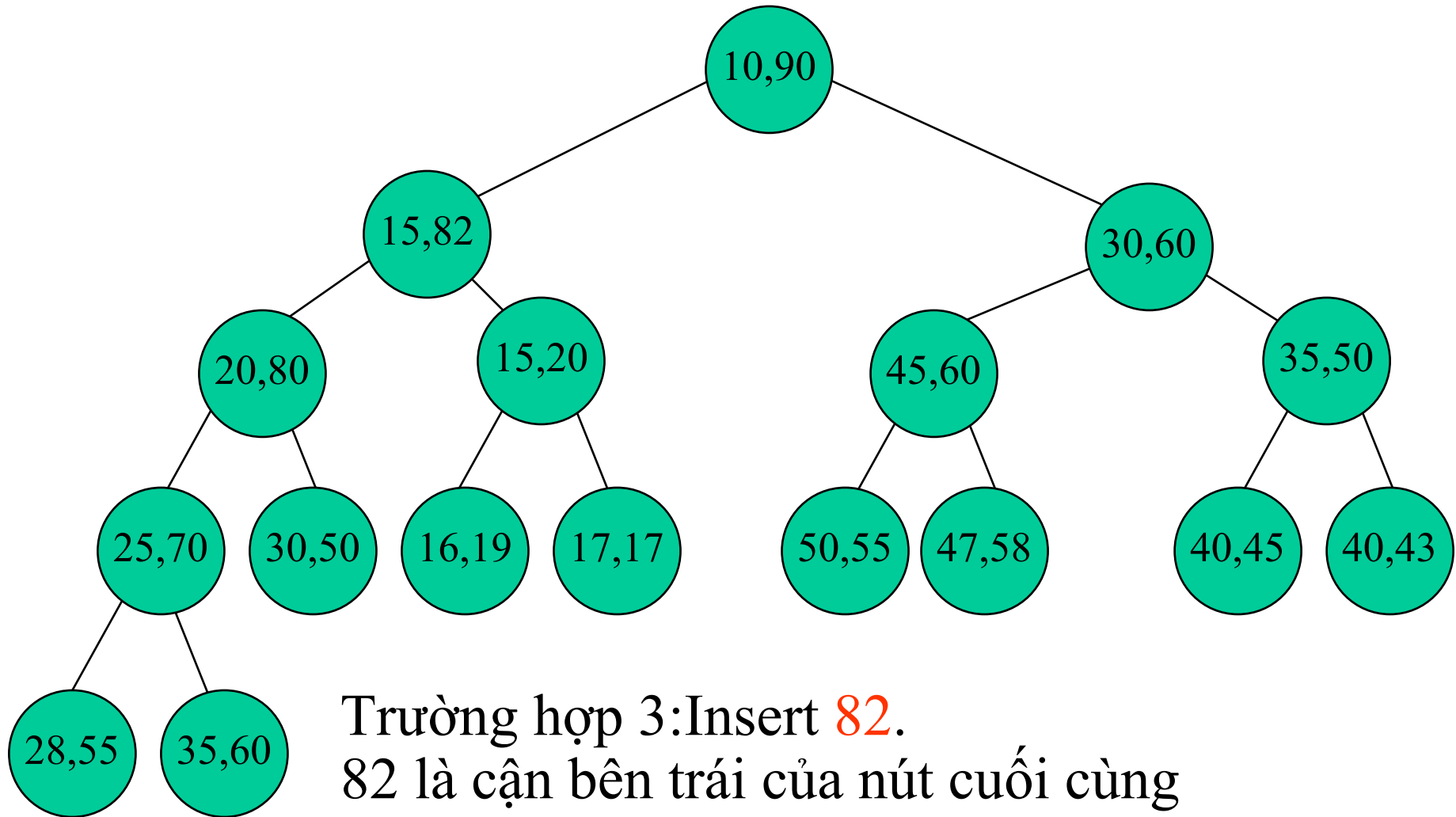
Giá trị 18 là cận bên trái của nút cuối cùng

Giá trị 18 này cũng được chèn vào min heap.

Yet Another Insert



Giá trị 82 phải đẩy dần lên trên để đảm bảo tính chất của max heap

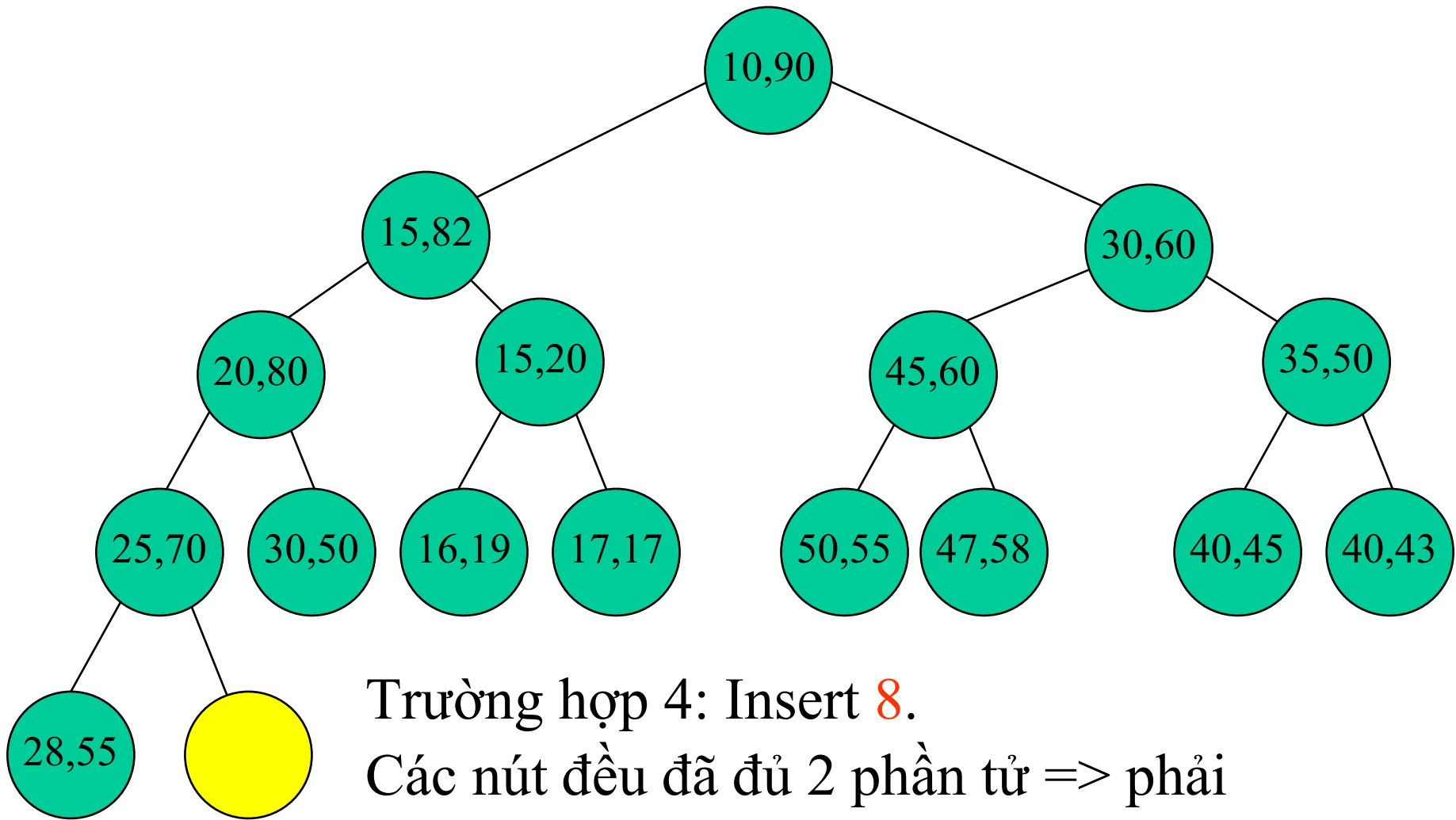


Trường hợp 3: Insert 82.

82 là cận bên trái của nút cuối cùng

82 được chèn vào max heap.

One More Insert Example

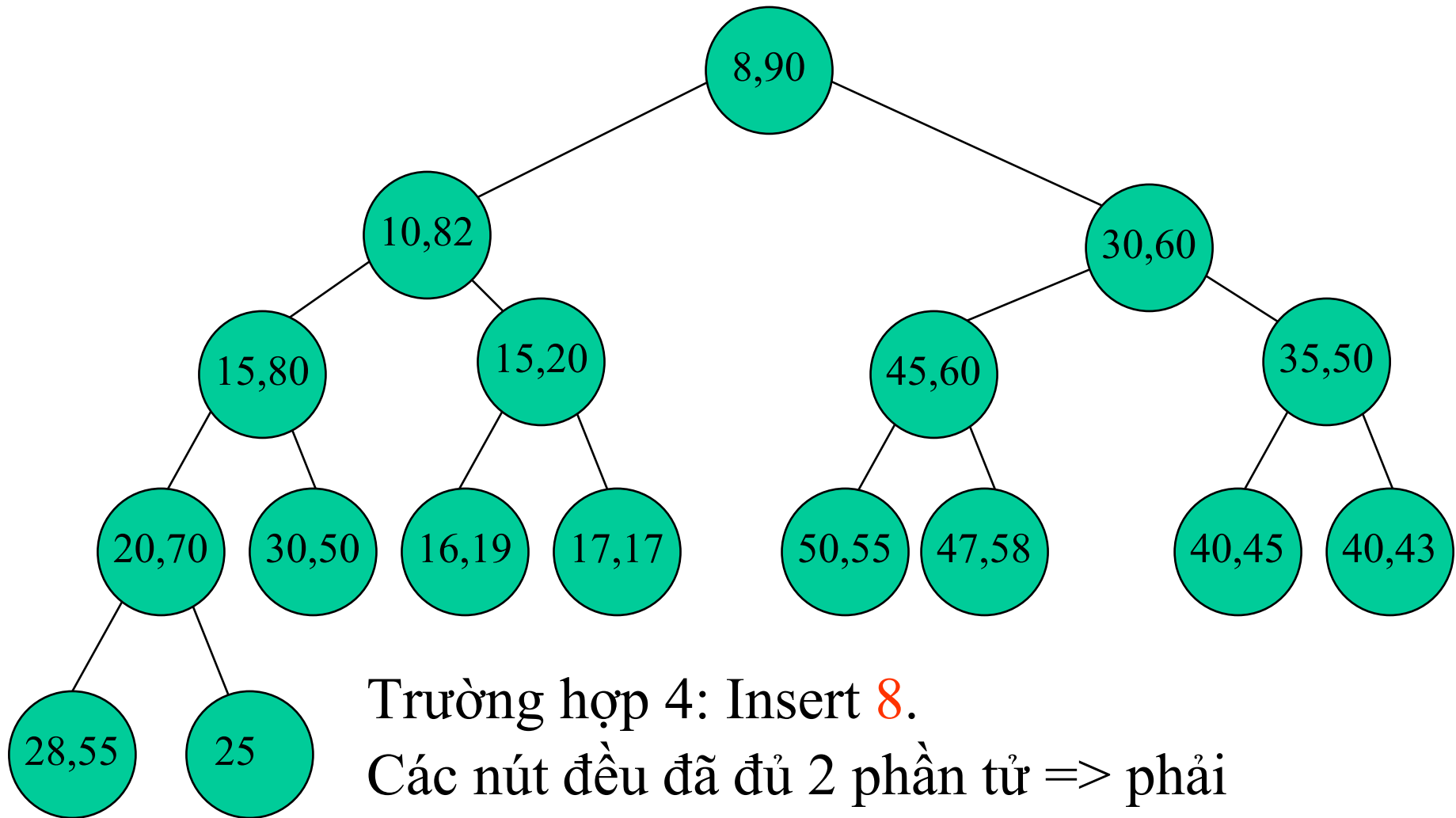


Trường hợp 4: Insert 8.

Các nút đều đã đủ 2 phần tử => phải tạo thêm một nút mới.

Giá trị 8 được chèn vào min heap.

Giá trị 18 phải đẩy dần lên trên để đảm bảo tính chất của min heap



Trường hợp 4: Insert 8.

Các nút đều đã đủ 2 phần tử => phải tạo thêm một nút mới.

Giá trị 8 được chèn vào min heap.

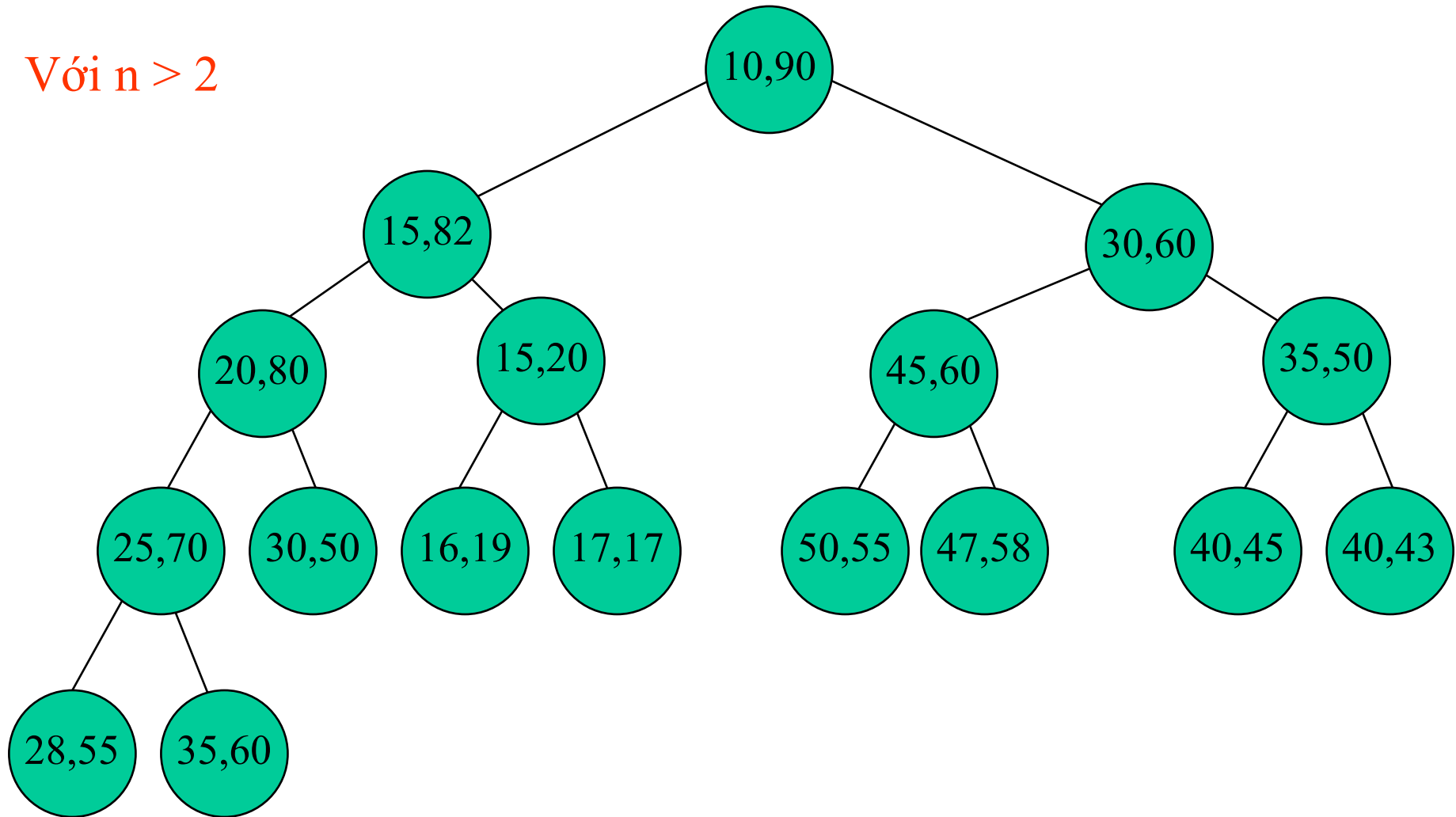
Remove Min Element

Gọi n là số phần tử của Interval Heap (mỗi nút có 2 phần tử)

- Nếu $n = 0 \Rightarrow$ heap rỗng, kết thúc.
- Nếu $n = 1 \Rightarrow$ sau khi remove heap becomes empty.
- Nếu $n = 2$ - heap chỉ có một nút \Rightarrow do đó để xóa min ta xóa cận trái.

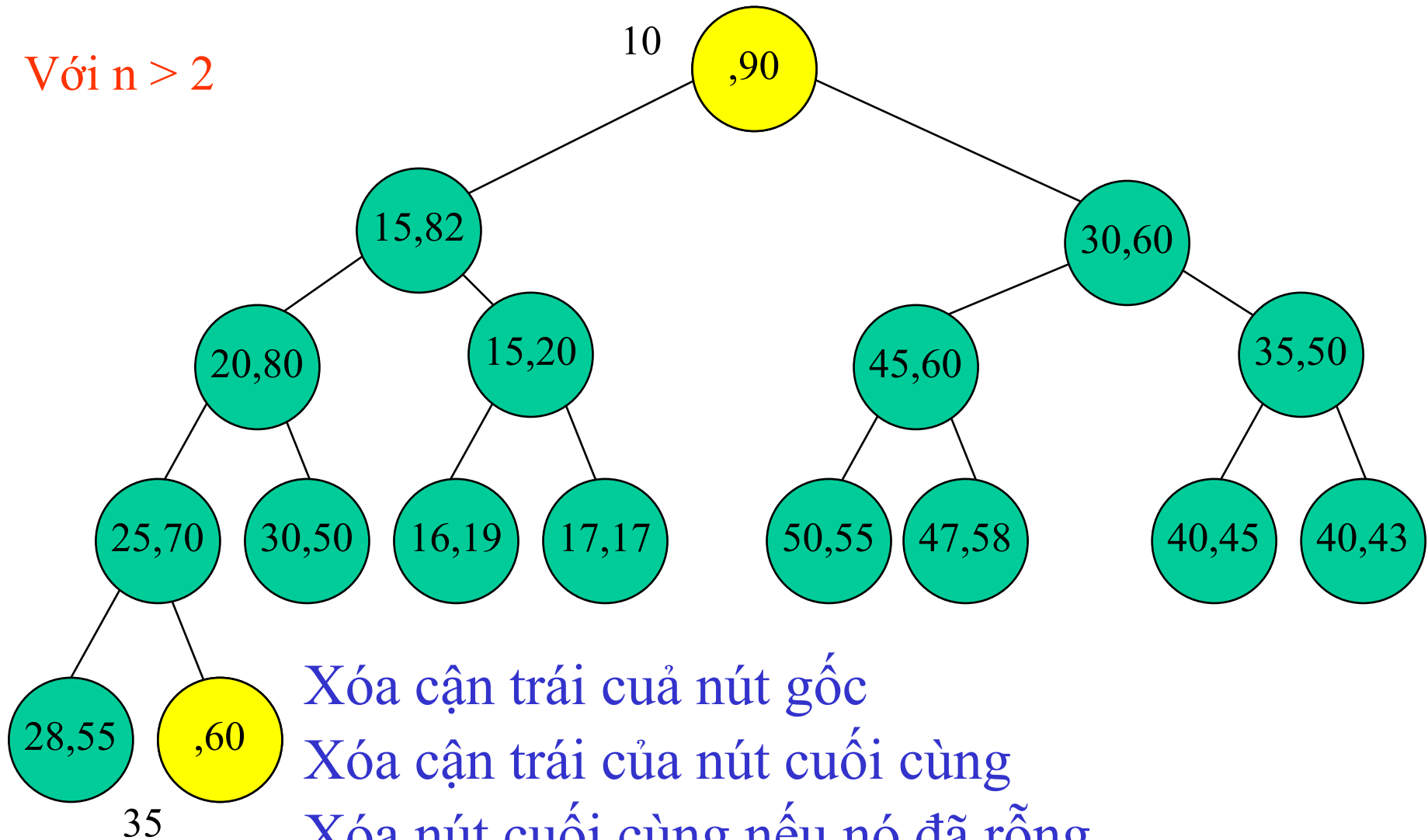
Remove Min Element

Với $n > 2$



Remove Min Element

Với $n > 2$



Xóa cận trái của nút gốc

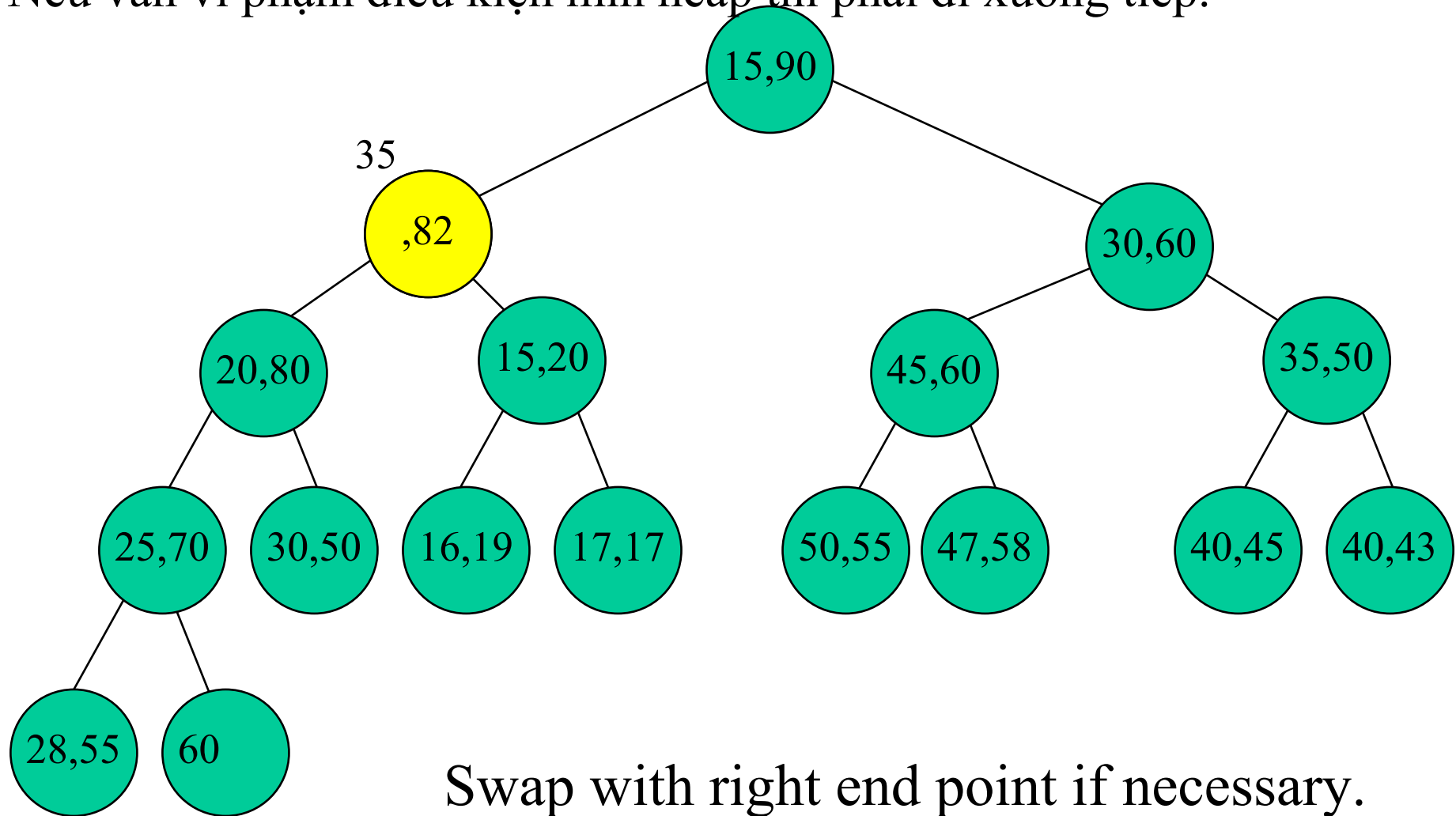
Xóa cận trái của nút cuối cùng

Xóa nút cuối cùng nếu nó đã rỗng

Chèn lại min heap bắt đầu từ nút gốc.

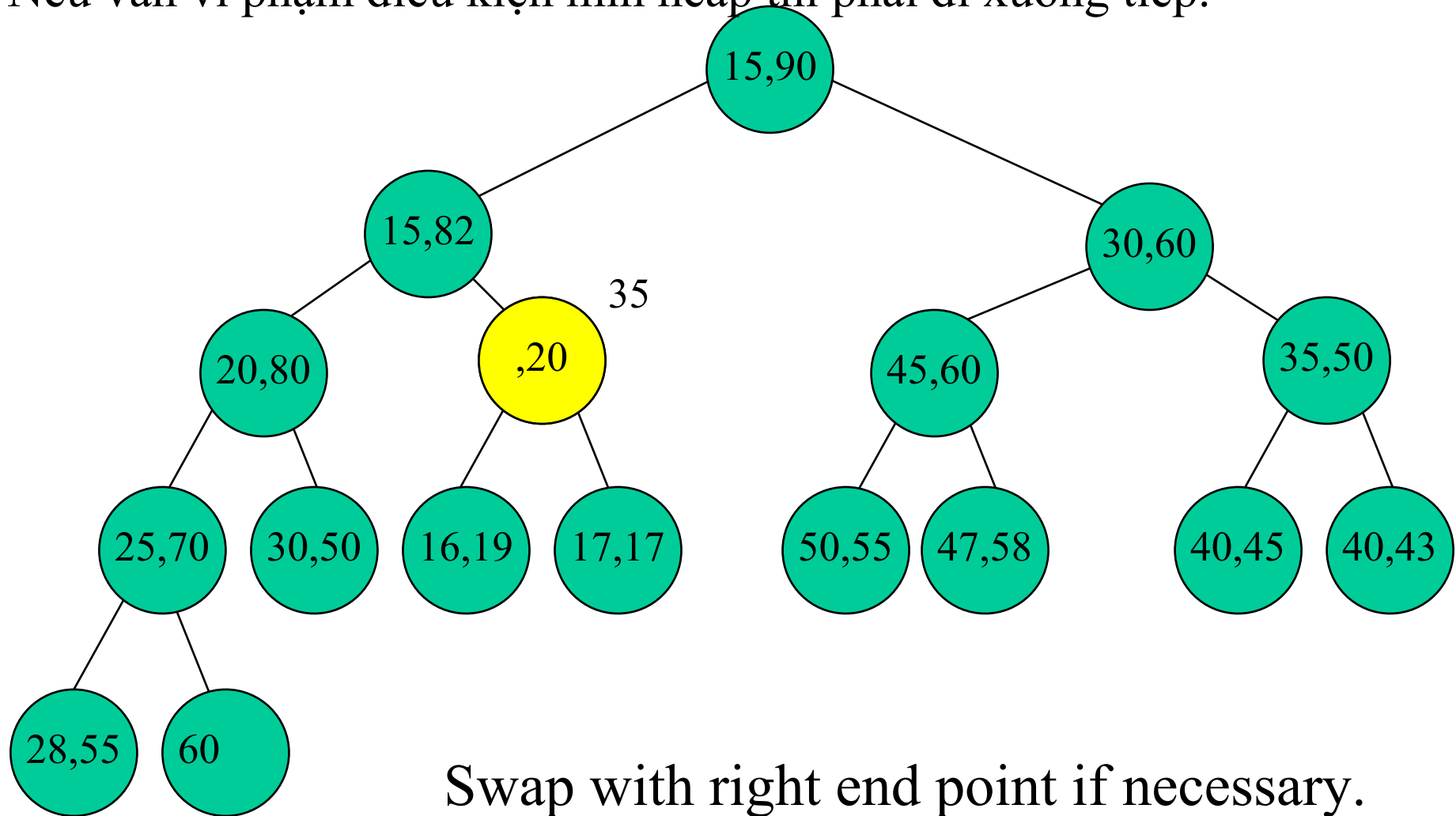
Đi xuống: trong 2 nút con trái, phải chọn nút có cận trái nhỏ hơn, đẩy giá trị cận trái này lên nút cha.

Nếu vẫn vi phạm điều kiện min heap thì phải đi xuống tiếp.



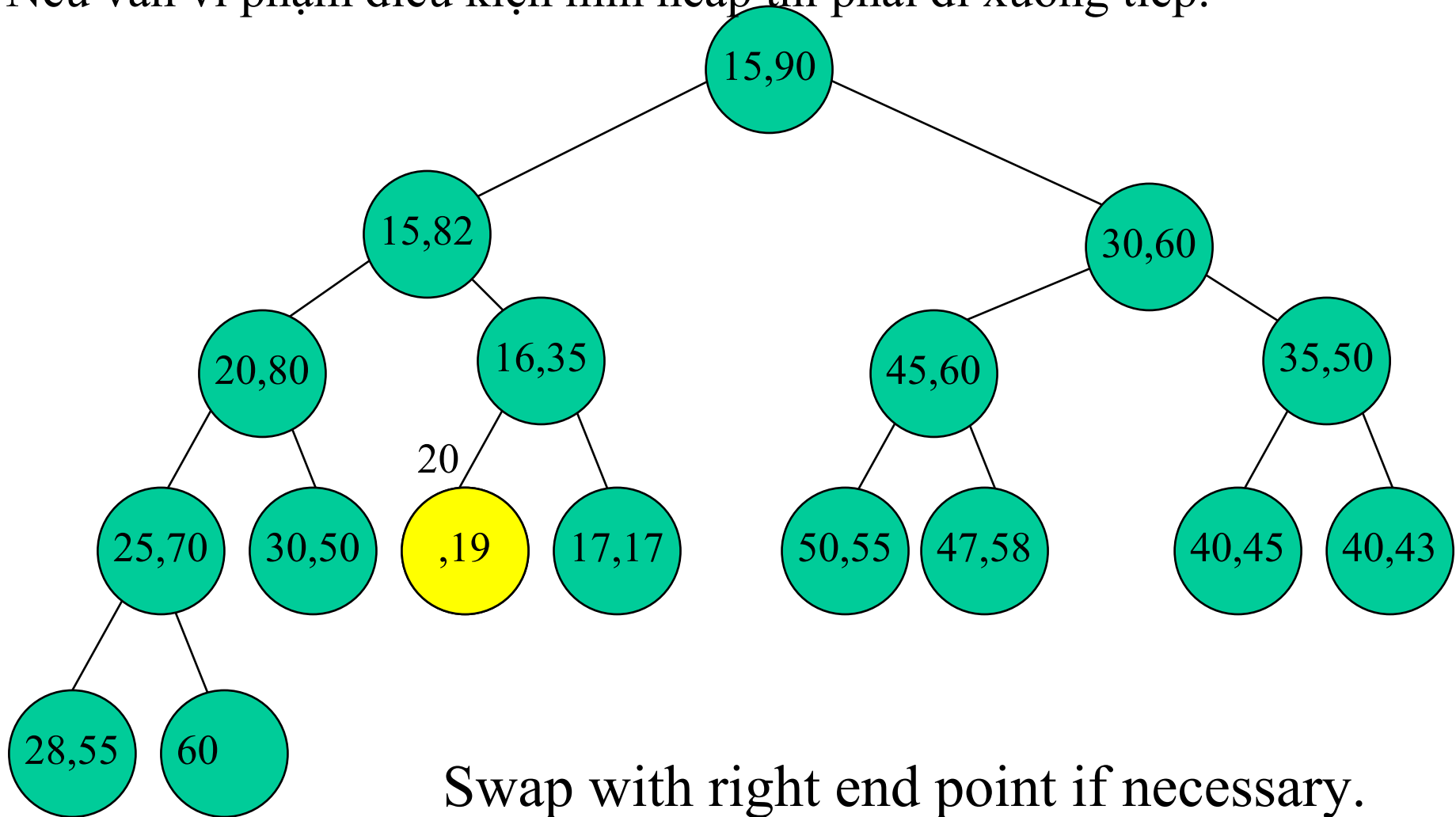
Đi xuống: trong 2 nút con trái, phải chọn nút có cận trái nhỏ hơn, đẩy giá trị cận trái này lên nút cha.

Nếu vẫn vi phạm điều kiện min heap thì phải đi xuống tiếp.



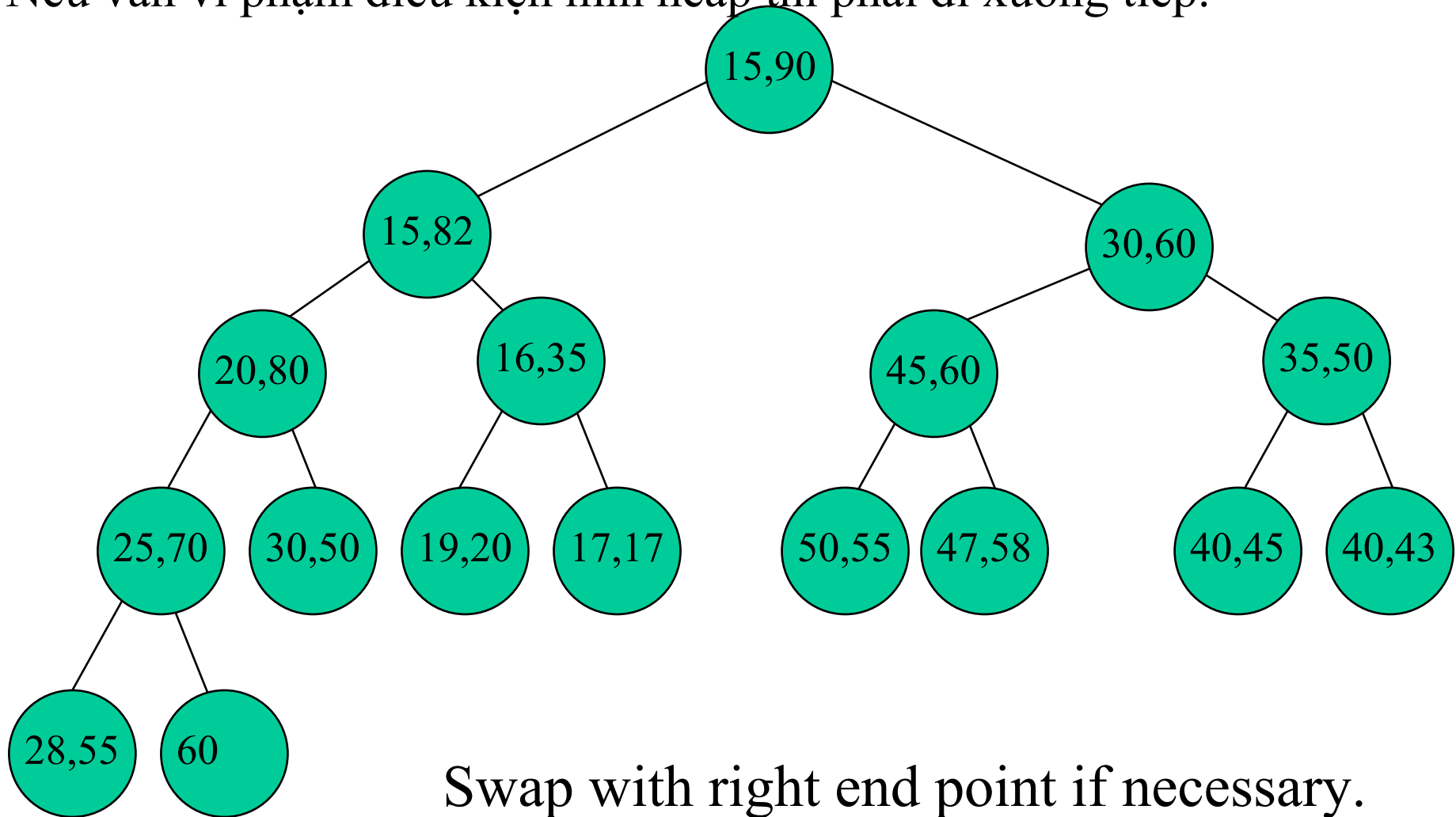
Đi xuống: trong 2 nút con trái, phải chọn nút có cận trái nhỏ hơn, đẩy giá trị cận trái này lên nút cha.

Nếu vẫn vi phạm điều kiện min heap thì phải đi xuống tiếp.



Đi xuống: trong 2 nút con trái, phải chọn nút có cận trái nhỏ hơn, đẩy giá trị cận trái này lên nút cha.

Nếu vẫn vi phạm điều kiện min heap thì phải đi xuống tiếp.



Application Of Interval Heaps

- Bài toán tìm kiếm khoảng bổ sung (complementary range search).
 - Giả sử có một tập hợp điểm 1 chiều (numbers - các con số).
 - $O(\log n)$
 - Thời gian để chèn vào một điểm (Insert a point).
 - $O(\log n)$
 - Remove a point.
 - $O(\log n)$
 - Tìm tất cả các điểm không thuộc đoạn $[a, b]$, $a \leq b$.
 - $O(k)$, với k là số điểm không thuộc khoảng.

Application Of Interval Heaps

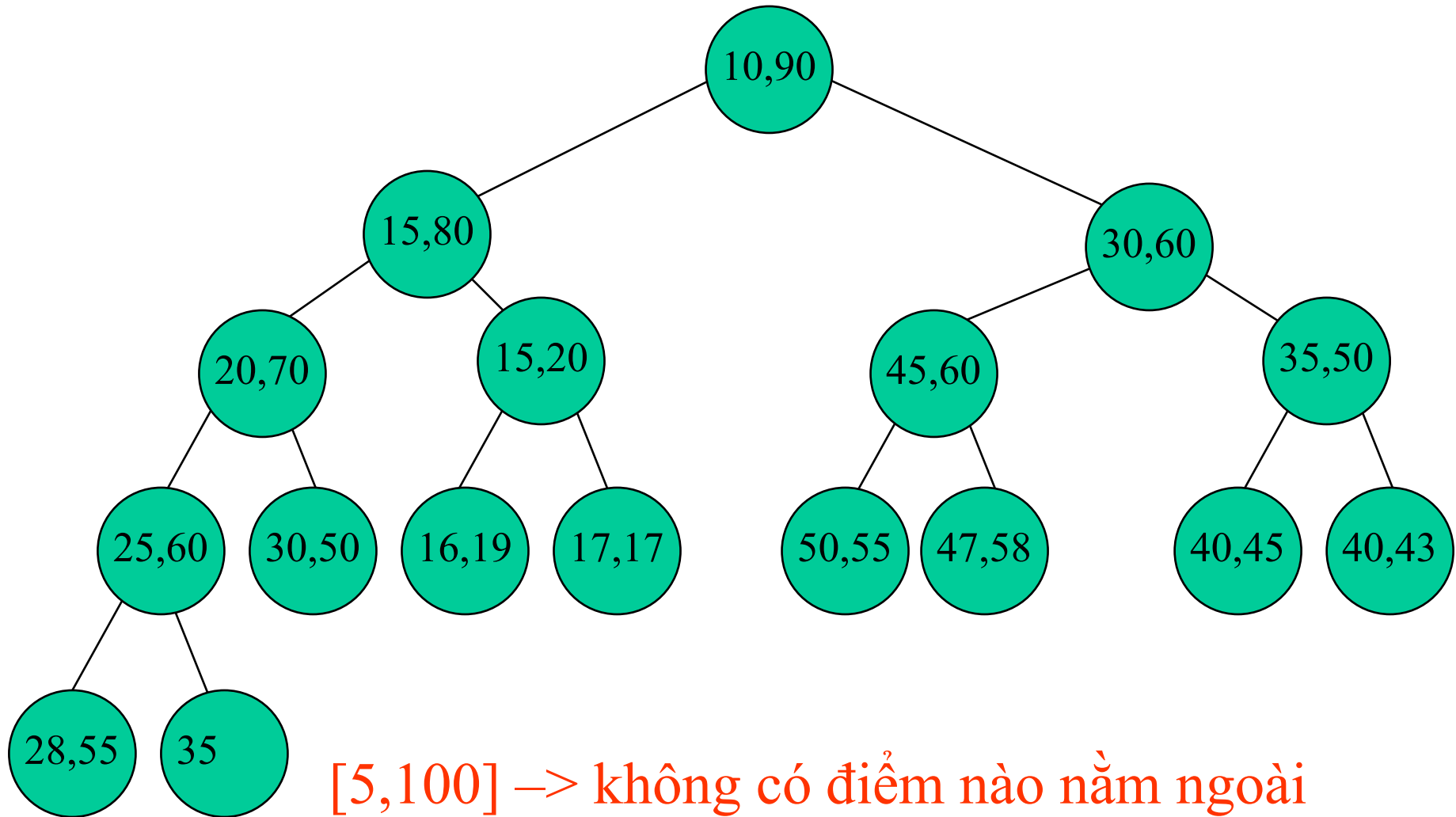
- Ví dụ nếu tập hợp điểm là:
 - 3, 4, 5, 6, 8, 12,
- Thì số điểm ngoài khoảng $[5,7]$ là: 3, 4, 8, 12.

Thuật toán tìm số điểm ngoài $[a,b]$

- Nếu khoảng của nút gốc chứa đoạn $[a,b]$, thì tất cả các điểm trong Interval Heap đều thuộc đoạn $[a,b]$ (như vậy không có điểm nào ngoài đoạn $[a,b]$), return.
- Ngược lại:
 - Thông báo điểm cận bên trái hoặc cận bên phải là điểm ngoài $[a,b]$
 - Tìm kiếm đệ qui xuống cây bên trái
 - Tìm kiếm đệ qui xuống cây bên phải

•

Thuật toán tìm số điểm ngoài $[a,b]$



$[5,100] \rightarrow$ không có điểm nào nằm ngoài

$[2,65] \rightarrow 90, 80, 70$