

# Árvore (Binária) de Pesquisa Balanceada

## Árvore de Intervalos

## Introdução

### Considere-se a seguinte situação

- Tem-se um conjunto de intervalos sobre o qual é necessário as seguintes operações de forma eficiente
  - inserir um intervalo
  - remover um intervalo
  - pesquisar um intervalo X: verificar se um dado X sobrepõe a um dos intervalos existentes

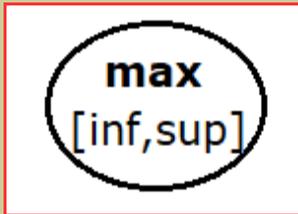
### Ideia

- Acrescentar a uma Árvore Binária de Pesquisa (ABP) um conjunto de intervalos, de forma que todas as operações tenham ordem de complexidade  $O(\log n)$
- A ABP pode ser não balanceada ou balanceada (como é o caso de uma Árvore Preto-Vermelho e Árvore AVL)

## Definição

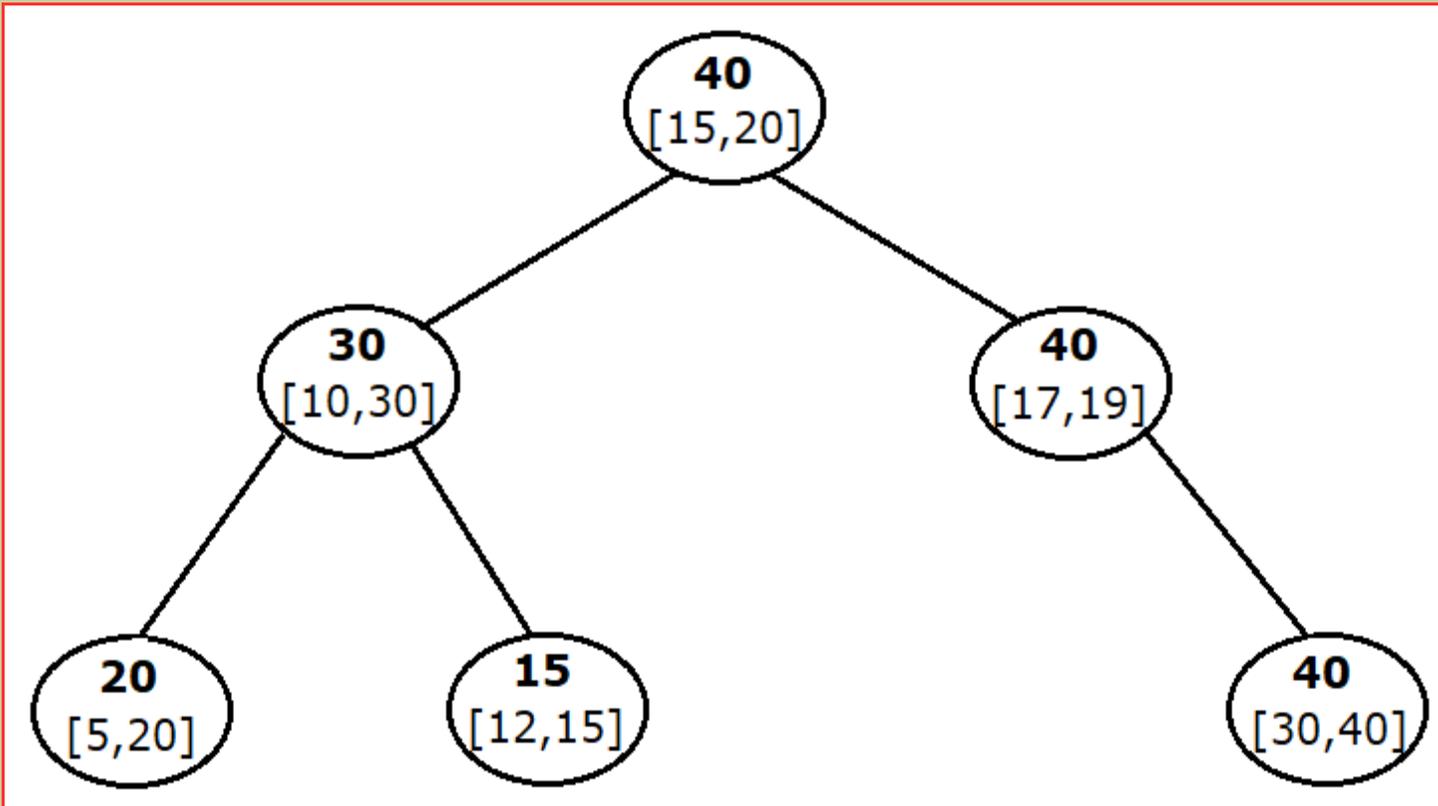
### Árvore de Intervalos

- É uma ABP em que cada nodo X guarda os seguintes dados
  - um intervalo que é representado como um par **[inf, sup]**
  - o maior limite superior dos intervalos da subárvore enraizada no nodo X, **max**



- O valor **inf** do intervalo é usado como **chave** para manter as propriedades de ABP
- As operações de inserção e remoção de um nodo seguem os mesmos métodos usados para as ABP

## Exemplo



- Informação do nodo (30; [10,30]):

- **max** = **30** (maior valor da subárvore com raiz neste nodo) =  $\max\{20, 15, 30\}$
- **inf** = **10** (limite inferior do intervalo e chave da ABP)
- **sup** = **30** (limite superior do intervalo)

## Operações sobre uma árvore de intervalos

### Inserir um intervalo numa árvore de intervalos

- É a mesma usada para as ABP, em que a chave é o valor de **inf**
- Um nodo com a informação de um intervalo [inf, sup] é criado da seguinte forma:
  - o intervalo [**inf**, **sup**] e o valor **max** = **sup**
- A inserção do intervalo [inf, sup] é feito da forma que se segue

Passo 1. criar o nodo com o intervalo [**inf**, **sup**] e o valor **max**, em que  $\text{max} = \text{sup}$

Passo 2. inserir o novo nodo da árvore, usando como **chave** o valor de **inf** dos nodos

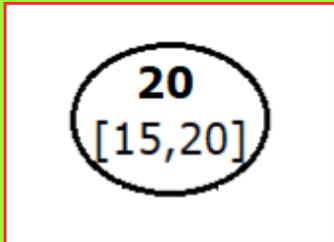
Passo 3. atualizar os valores de **max** dos antepassados do nodo inserido

## Inserir um intervalo numa árvore de intervalos

- Exemplo: inserir o intervalo  $[15, 20]$  numa árvore vazia

Passo 1. criar o nodo (**20**,  $[15, 20]$ ), em que o valor **max** = sup = **20**

Passo 2. como é o primeiro nodo inserido, então é a raiz da árvore



Passo 3. como é o primeiro nodo inserido, a atualização não é necessária

## Inserir um intervalo numa árvore de intervalos

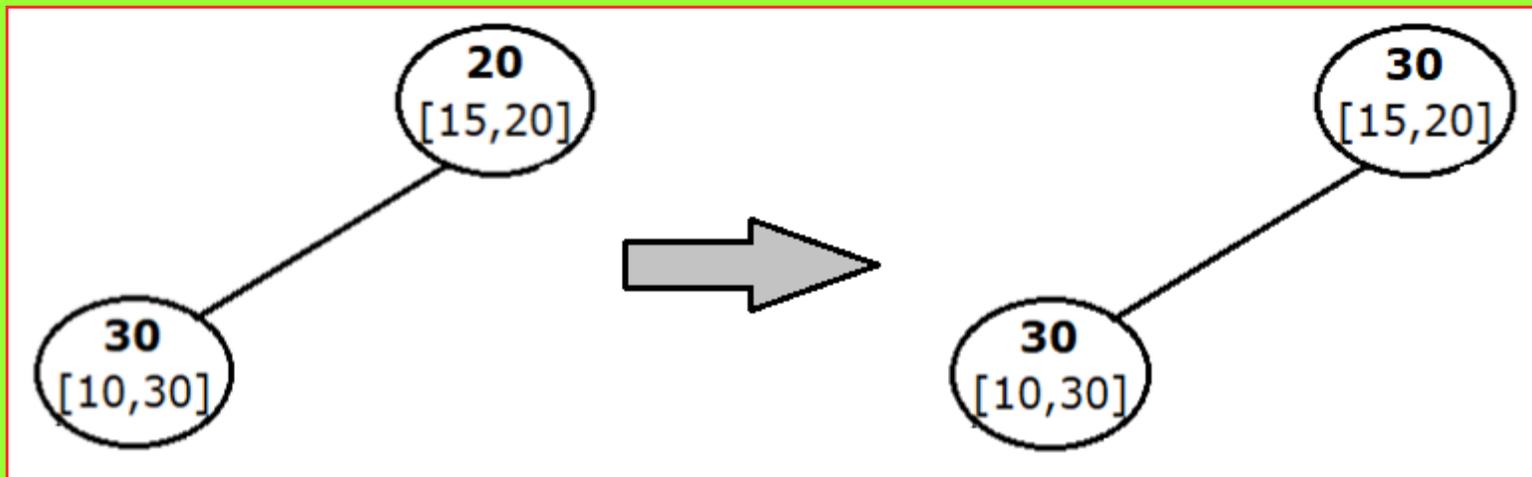
- Exemplo: inserir o intervalo  $[10, 30]$  na árvore anterior

Passo 1. criar o nodo  $(30, [10, 30])$ , em que o valor **max** = sup = **30**

Passo 2. comparar o **inf** deste nodo com o **inf** da raiz:

(se for menor, então inserir o nodo à esquerda, senão inserir à direita da raiz)

$10 < 15 \Rightarrow$  inserir à esquerda da raiz



Passo 3. atualizar o valor de **max** da raiz:

**max** =  $\max\{20, 30\} = 30$

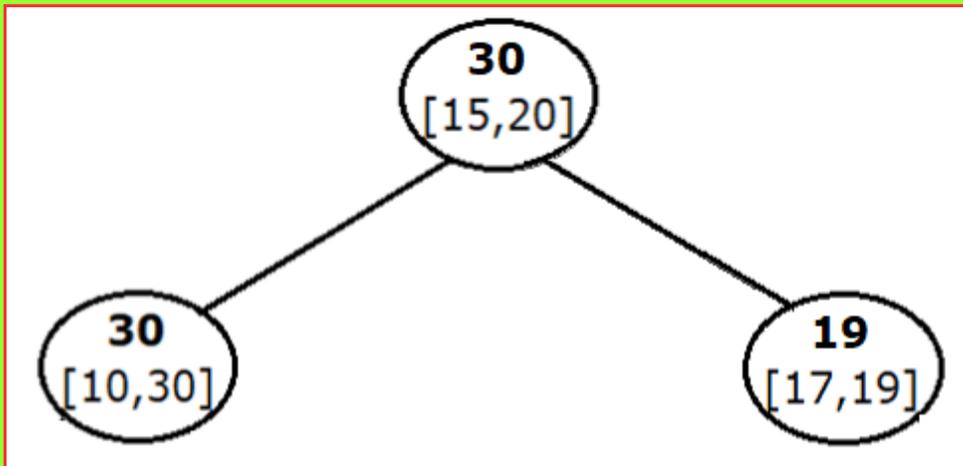
## Inserir um intervalo numa árvore de intervalos

- Exemplo: inserir o intervalo  $[17, 19]$  na árvore anterior

Passo 1. criar o nodo (**19**,  $[17, 19]$ ), em que o valor **max** = sup = **19**

Passo 2. comparar os valores **inf** do nodo e da raiz:

$17 > 15 \Rightarrow$  inserir à direita da raiz



Passo 3. atualizar o valor de **max** da raiz:

**max** =  $\max\{30, 30, 19\} = 30$  (sem alteração)

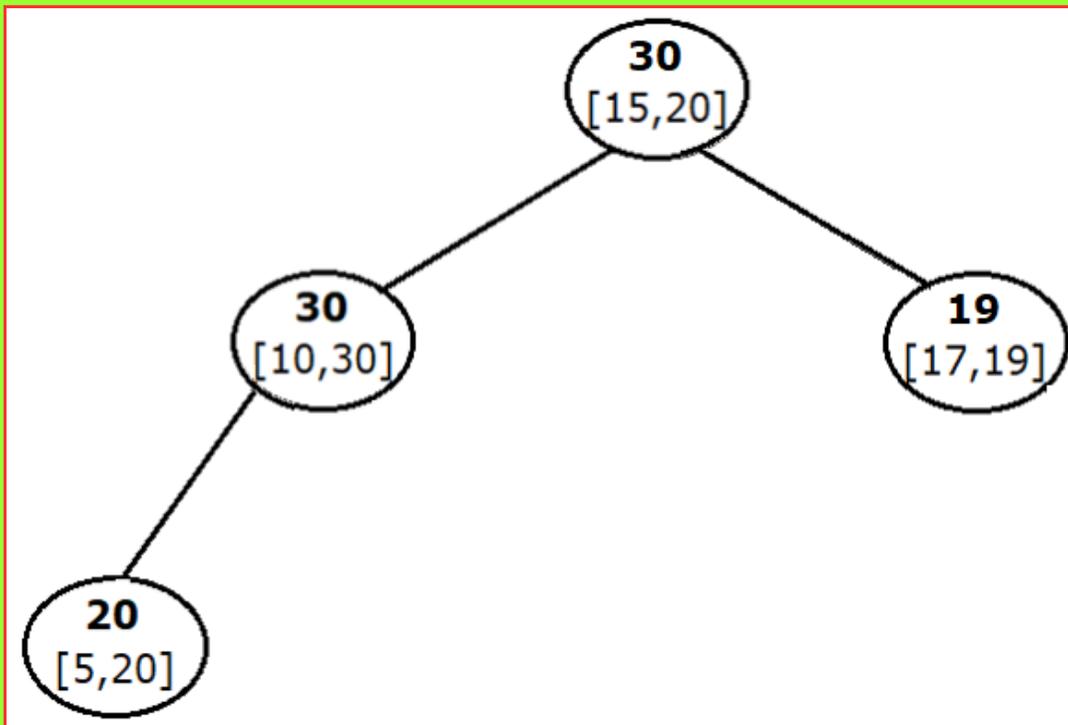
## Inserir um intervalo numa árvore de intervalos

- Exemplo: inserir o intervalo  $[5, 20]$  na árvore anterior

Passo 1. criar o nodo (**20**,  $[5, 20]$ ), em que o valor **max** = sup = **20**

Passo 2. comparar o **inf** do nodo e da raiz:  $5 < 15 \Rightarrow$  inserir à esquerda

Passo 2. comparar o **inf** do nodo e do filho esquerdo da raiz:  $5 < 10 \Rightarrow$  inserir à esquerda



Passo 3. atualizar os valores de **max** dos antepassados do nodo inserido: sem alteração

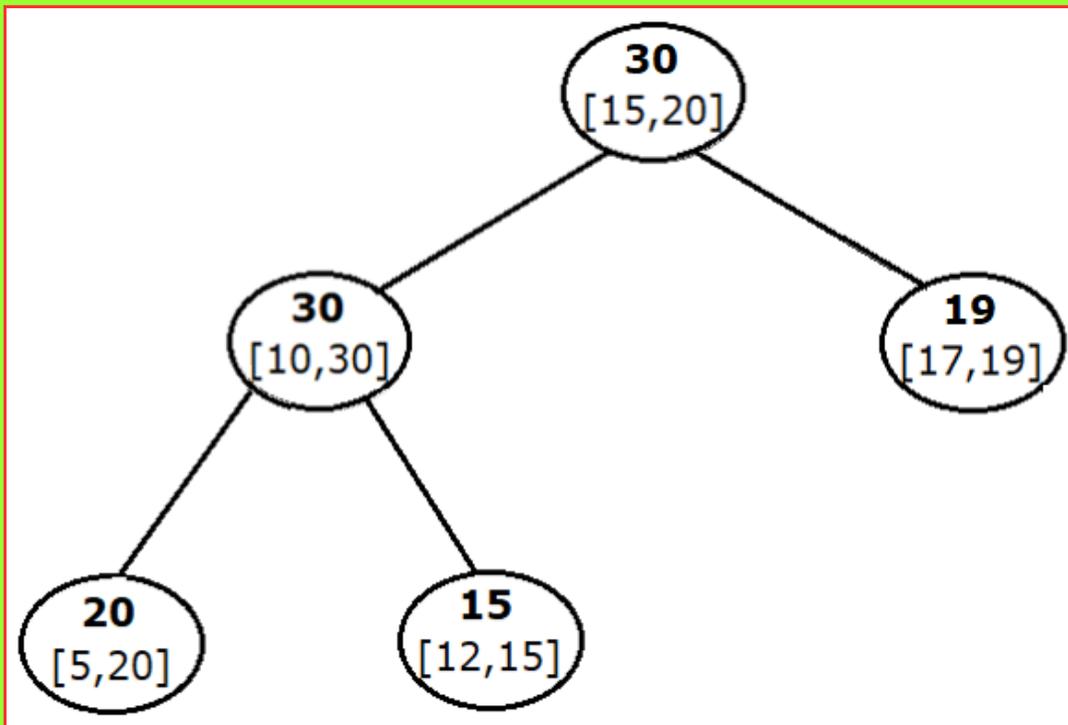
## Inserir um intervalo numa árvore de intervalos

- Exemplo: inserir o intervalo  $[12, 15]$  na árvore anterior

Passo 1. criar o nodo (**15**,  $[12, 15]$ ), em que o valor **max** = sup = **15**

Passo 2. comparar o **inf** do nodo e da raiz:  $12 < 15 \Rightarrow$  inserir à esquerda

Passo 2. comparar o **inf** do nodo e do filho esquerdo da raiz:  $12 > 10 \Rightarrow$  inserir à direita



Passo 3. atualizar os valores de **max** dos antepassados do nodo inserido: sem alteração

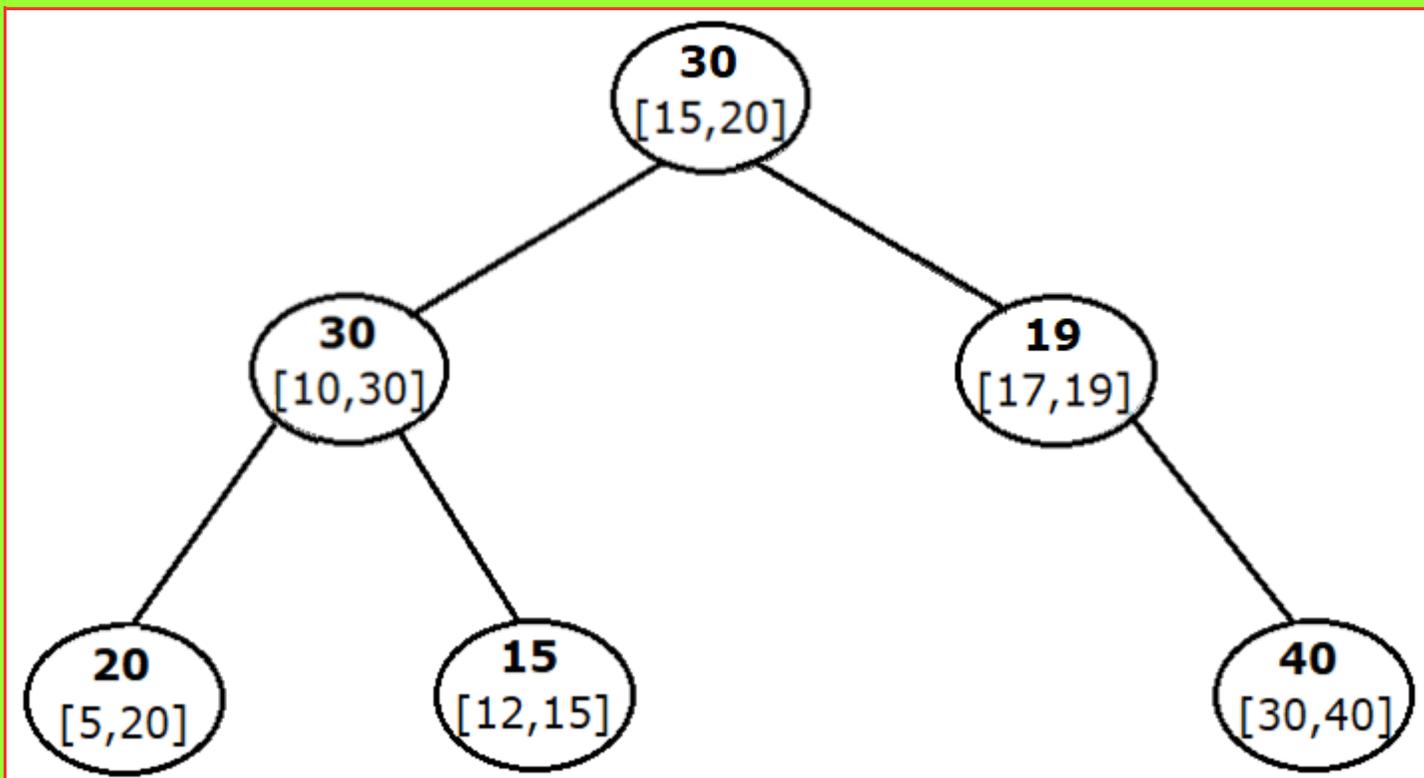
## Inserir um intervalo numa árvore de intervalos

- Exemplo: inserir o intervalo  $[30, 40]$  na árvore anterior

Passo 1. criar o nodo  $(40, [30, 40])$ , em que o valor **max** = sup = **40**

Passo 2. comparar o **inf** do nodo e da raiz:  $30 > 15 \Rightarrow$  inserir à direita da raiz

Passo 2. comparar o **inf** do nodo e do filho esquerdo da raiz:  $30 > 17 \Rightarrow$  inserir à direita

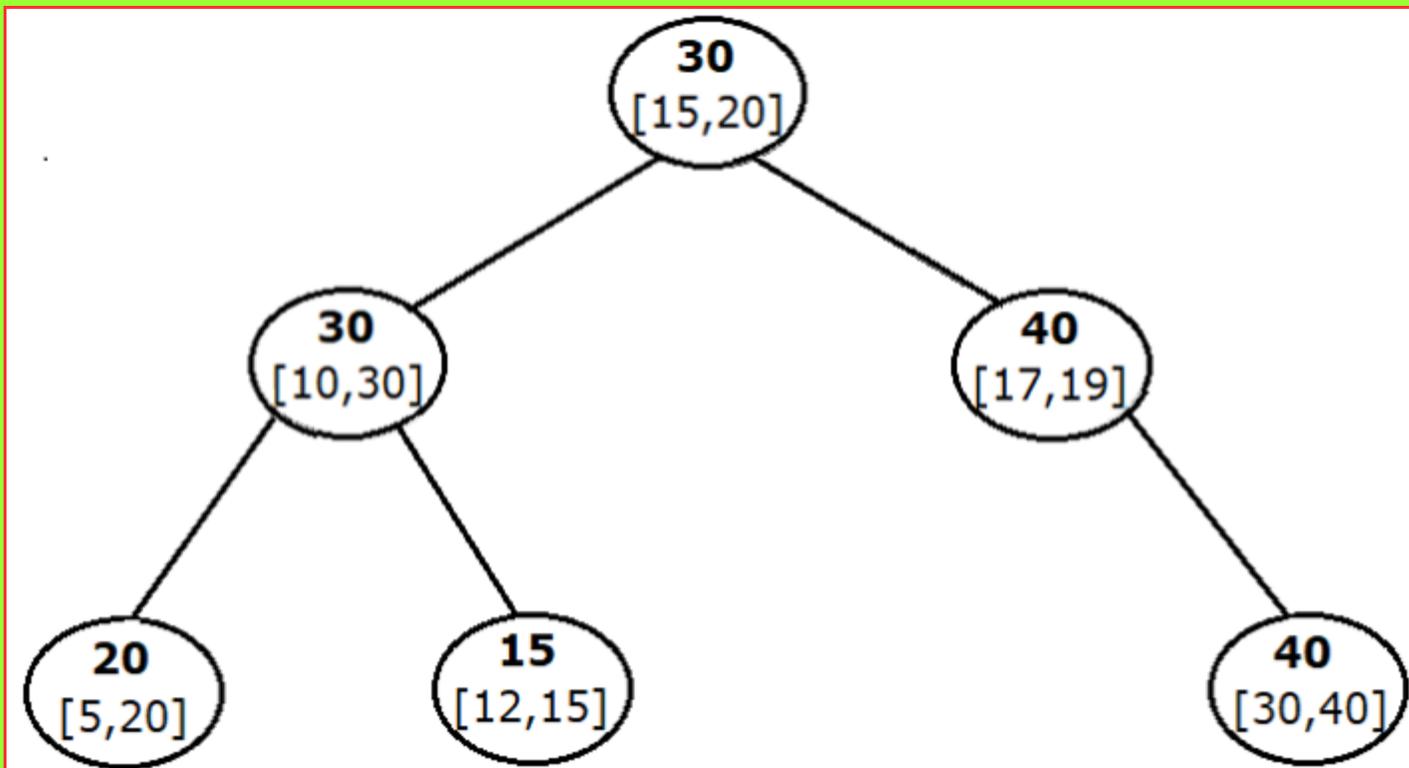


## Inserir um intervalo numa árvore de intervalos

- Exemplo: inserir o intervalo [30, 40] na árvore anterior

Passo 3. atualizar os valores de **max** dos antepassados do nodo inserido:

$$\mathbf{max}([17, 19]) = \max\{19, 40\} = \mathbf{40}$$

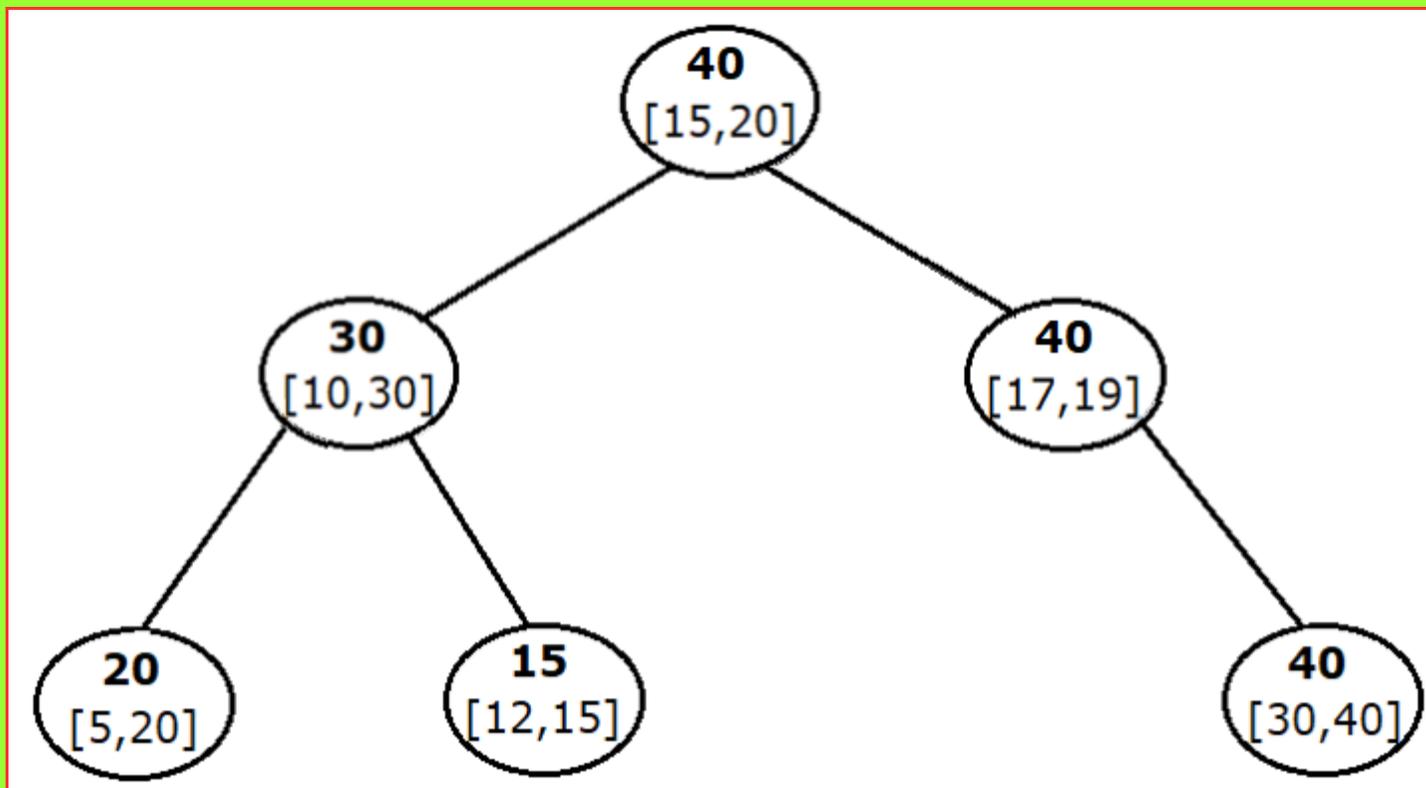


## Inserir um intervalo numa árvore de intervalos

- Exemplo: inserir o intervalo  $[30, 40]$  na árvore anterior

Passo 3. atualizar os valores de **max** dos antepassados do nodo inserido:

$$\mathbf{max}([15, 20]) = \max\{30, 30, 40\} = \mathbf{40}$$



## Pesquisar um intervalo numa árvore de intervalos

- É a principal operação sobre uma árvore de intervalos
- Pretende-se procurar um nodo da árvore cujo intervalo se sobrepõe ao intervalo a pesquisar

## Pesquisar um intervalo numa árvore de intervalos

- Algoritmo

Entrada: um intervalo X e uma árvore de intervalos T

Saída: o ponteiro para um nodo de T cujo intervalo se sobrepõe ao intervalo X ou NULL

**algoritmo** pesquisaIntervalo (T, X)

**se** (X sobrepõe-se ao intervalo de T) **então**

**devolver** T (nodo cujo intervalo se sobrepõe a X)

**fim\_se**

**se** (filho esquerdo de T  $\neq$  vazio **e** **inf** de X < **max** de filho esquerdo de T) **então**

**pesquisaIntervalo**(T→Esquerda, X) (pesquisar X no filho esquerdo de T)

**senão**

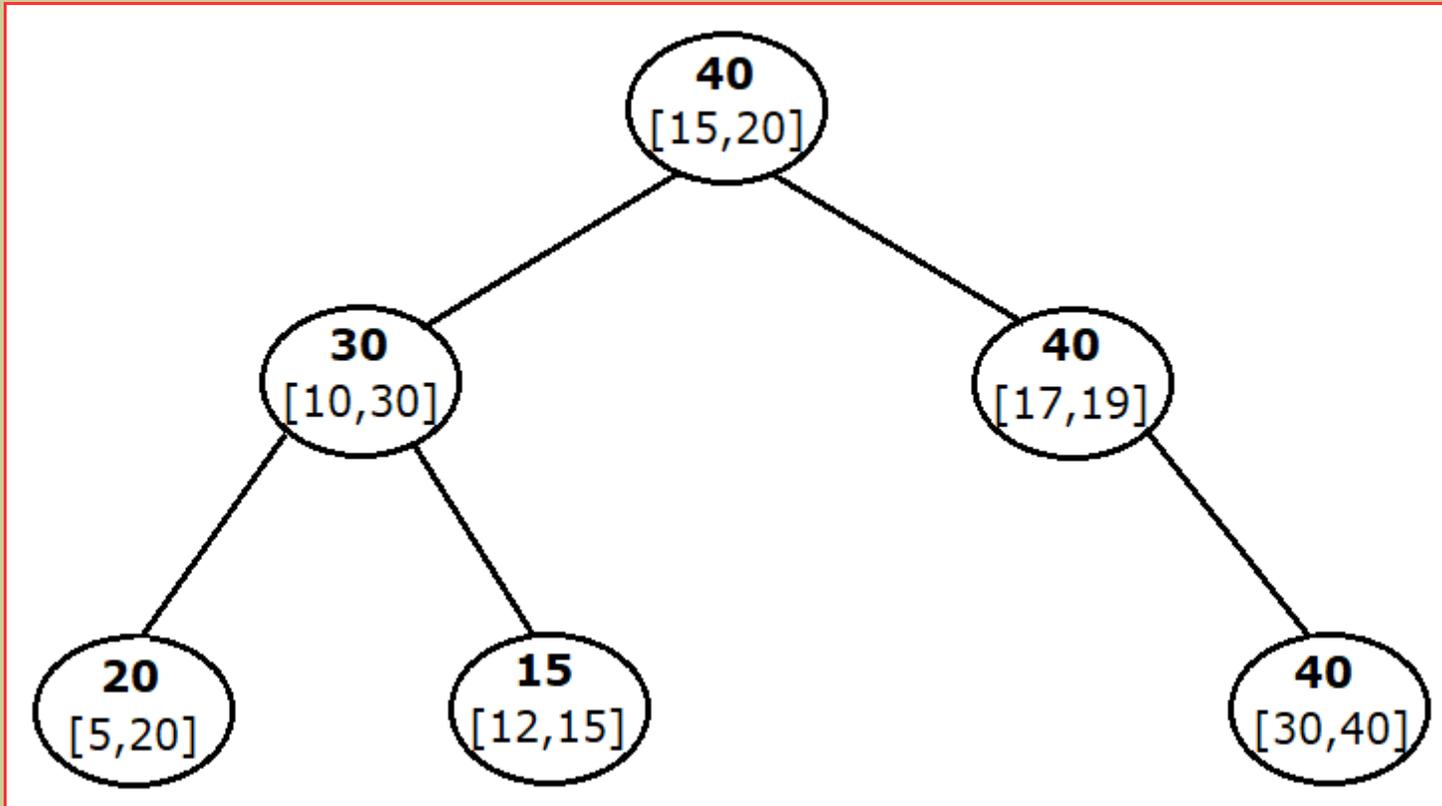
**pesquisaIntervalo**(T→Direita, X) (pesquisar X no filho direito de T)

**fim\_se**

**fim\_algoritmo**

## Pesquisar um intervalo numa árvore de intervalos

- Exemplo: procurar o intervalo  $[6, 7]$  na árvore em baixo



## Pesquisar um intervalo numa árvore de intervalos

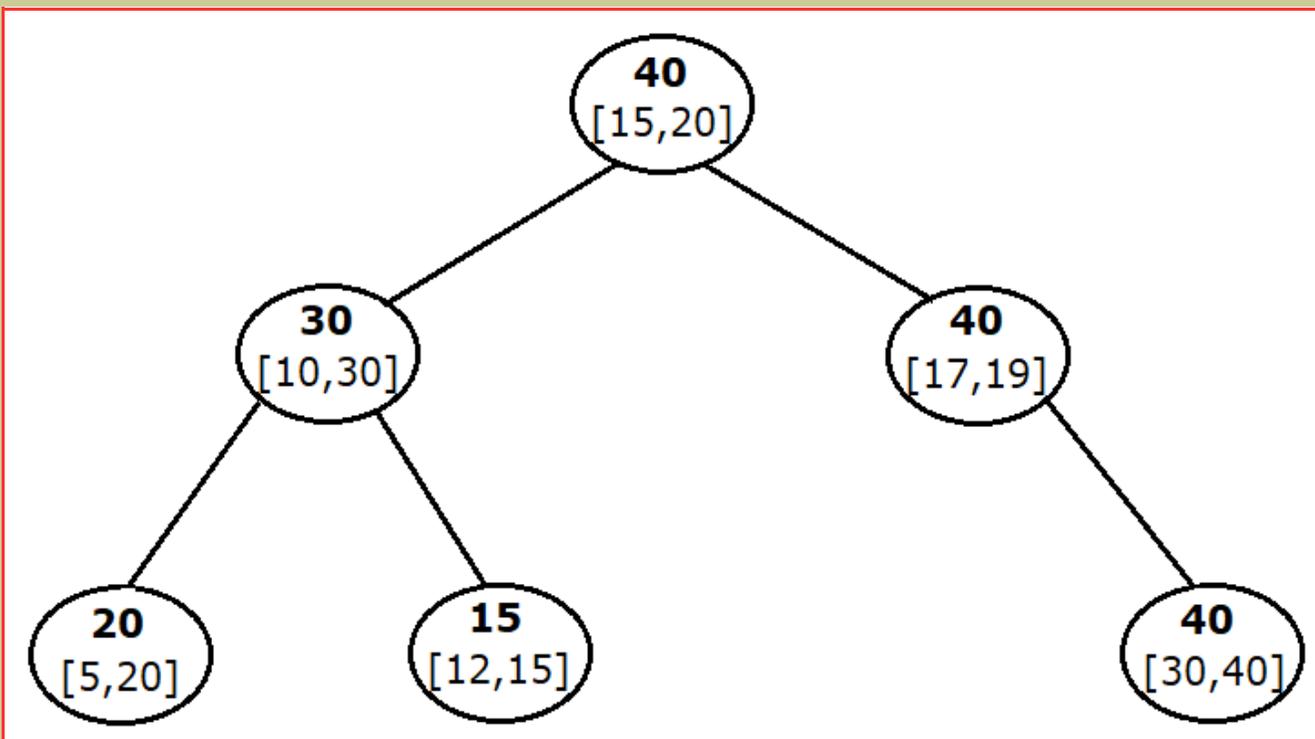
- Exemplo: procurar o intervalo  $[6, 7]$  na árvore T em baixo

**pesquisaIntervalo (T,  $[6,7]$ )**

$[6,7]$  **não sobrepõe** a  $[15,20]$  (intervalo de T)

filho esquerdo de T ( $[10,30]$ )  $\neq \emptyset$  e  $\mathbf{inf}([6,7])=6 < \mathbf{max}([10,30])=30$

pesquisaIntervalo (T $[10,30]$ ,  $[6,7]$ )



## Pesquisar um intervalo numa árvore de intervalos

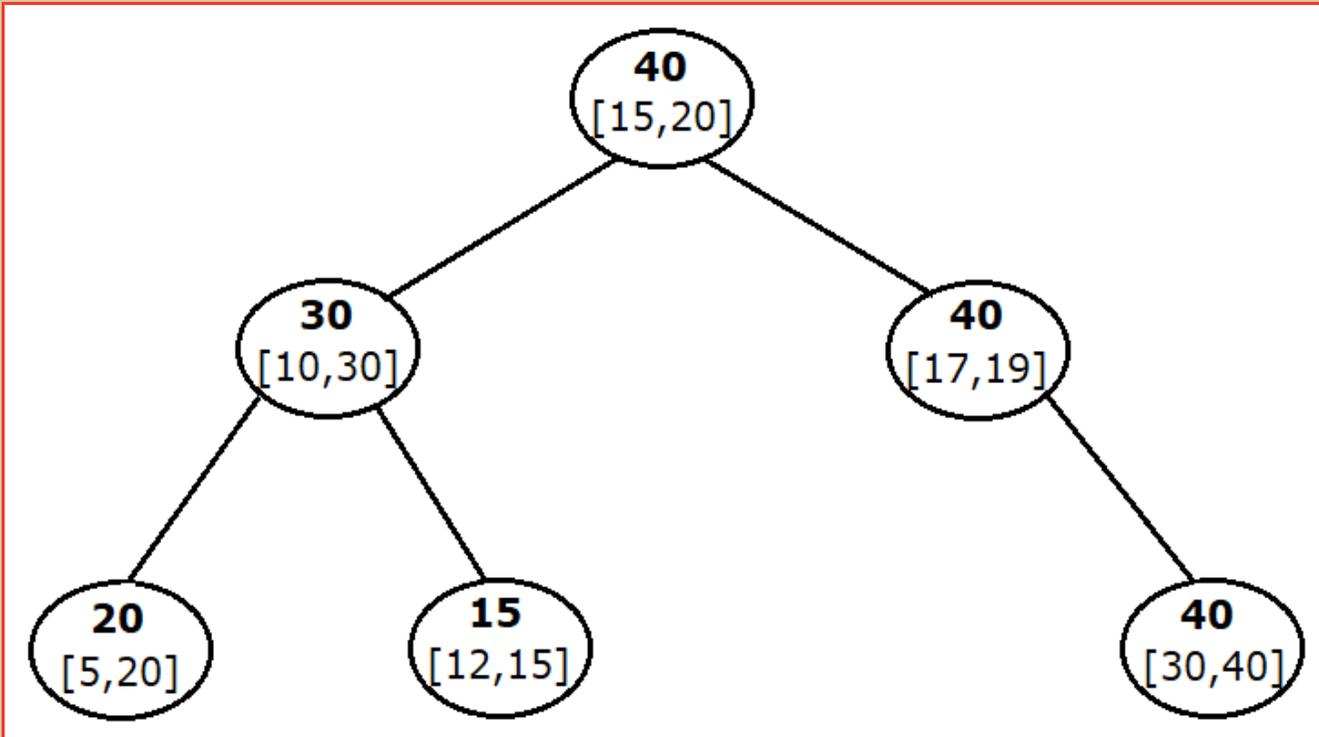
- Exemplo: procurar o intervalo  $[6, 7]$  na árvore T em baixo

**pesquisaIntervalo (T[10,30], [6,7])**

$[6,7]$  **não sobrepõe** a  $[10,30]$  (intervalo de T[10,30])

filho esquerdo de T[10,30] ( $[5,20]$ )  $\neq \emptyset$  e **inf**( $[6,7]$ )=6 < **max**( $[5,20]$ )=20

pesquisaIntervalo (T[5,20], [6,7])



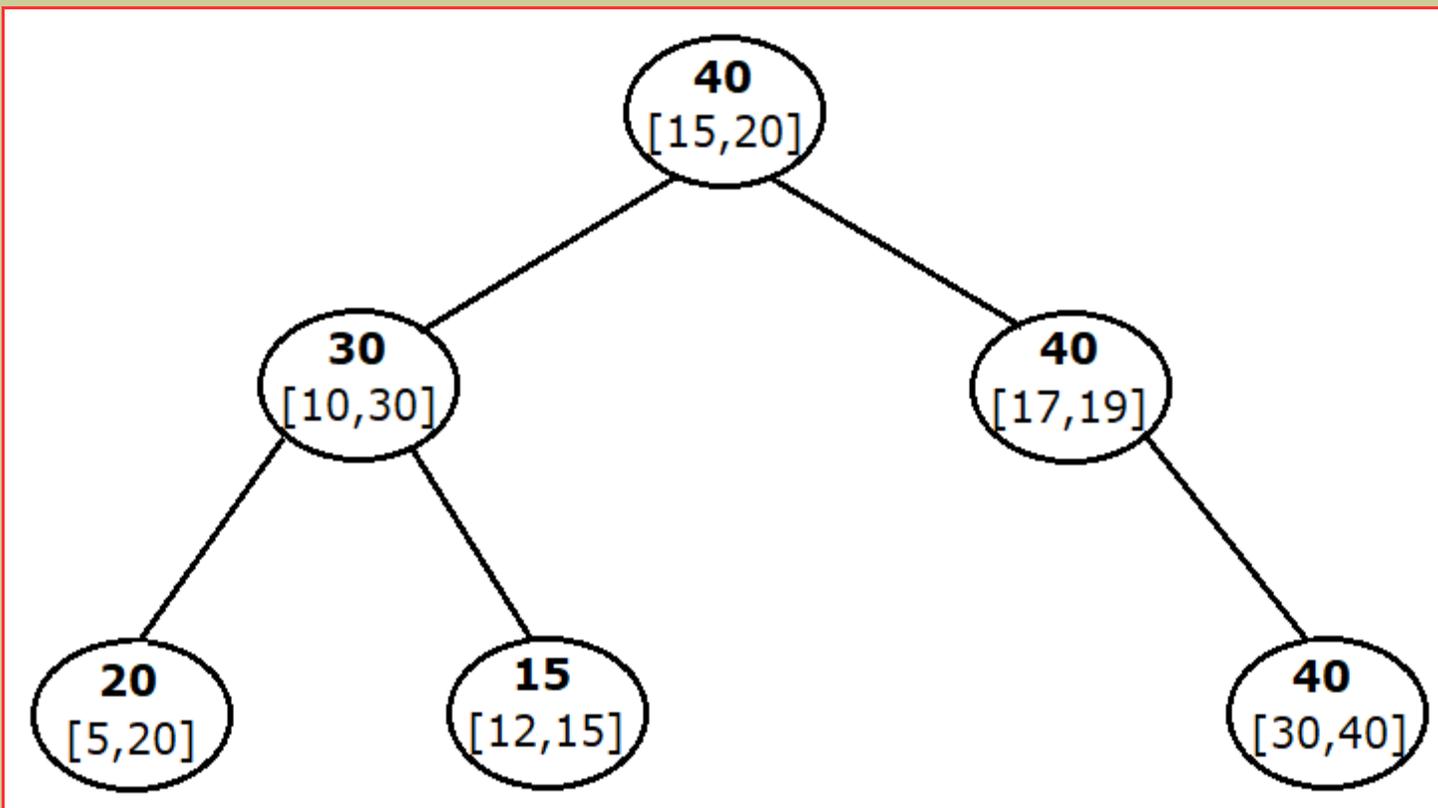
## Pesquisar um intervalo numa árvore de intervalos

- Exemplo: procurar o intervalo  $[6, 7]$  na árvore  $T$  em baixo

**pesquisaIntervalo ( $T[5,20]$ ,  $[6,7]$ )**

$[6,7]$  **sobrepe** a  $[5,20]$  (intervalo de  $T[5,20]$ )

**devolver**  $[5,20]$

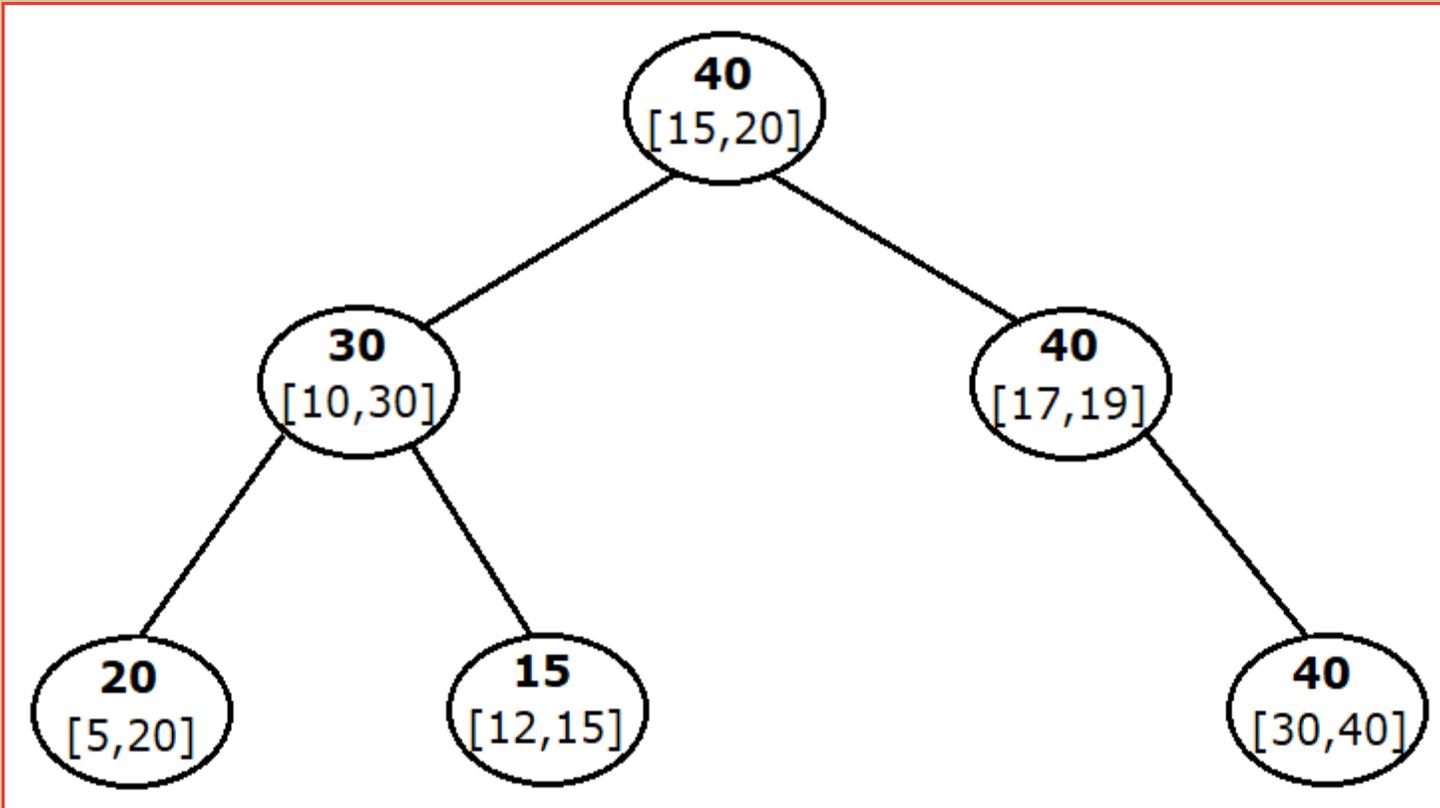


## Remover um intervalo numa árvore de intervalos

- Usa o mesmo método que uma ABP usa para remover um nodo (a chave é o **inf**)
- O método é o seguinte:
  - pesquisar pelo nodo da árvore cujo intervalo se sobrepõe ao intervalo a remover (operação pesquisarIntervalo)
  - verificar se o nodo encontrado tem filhos:
    - se não tem filhos (é uma folha), então basta remover este nodo
    - se tem apenas o filho esquerdo, então substituir o nodo pelo filho esquerdo (ligar o pai do nodo ao filho esquerdo do nodo)
    - se tem apenas o filho direito, então substituir o nodo pelo filho direito (ligar o pai do nodo ao filho direito do nodo)
    - se tem dois filhos, então substituir o nodo pelo descendente
      - mais à direita do seu filho esquerdo, ou
      - mais à esquerda do seu filho direito
  - atualizar os valores de **max** dos nodos antecessores do nodo removido

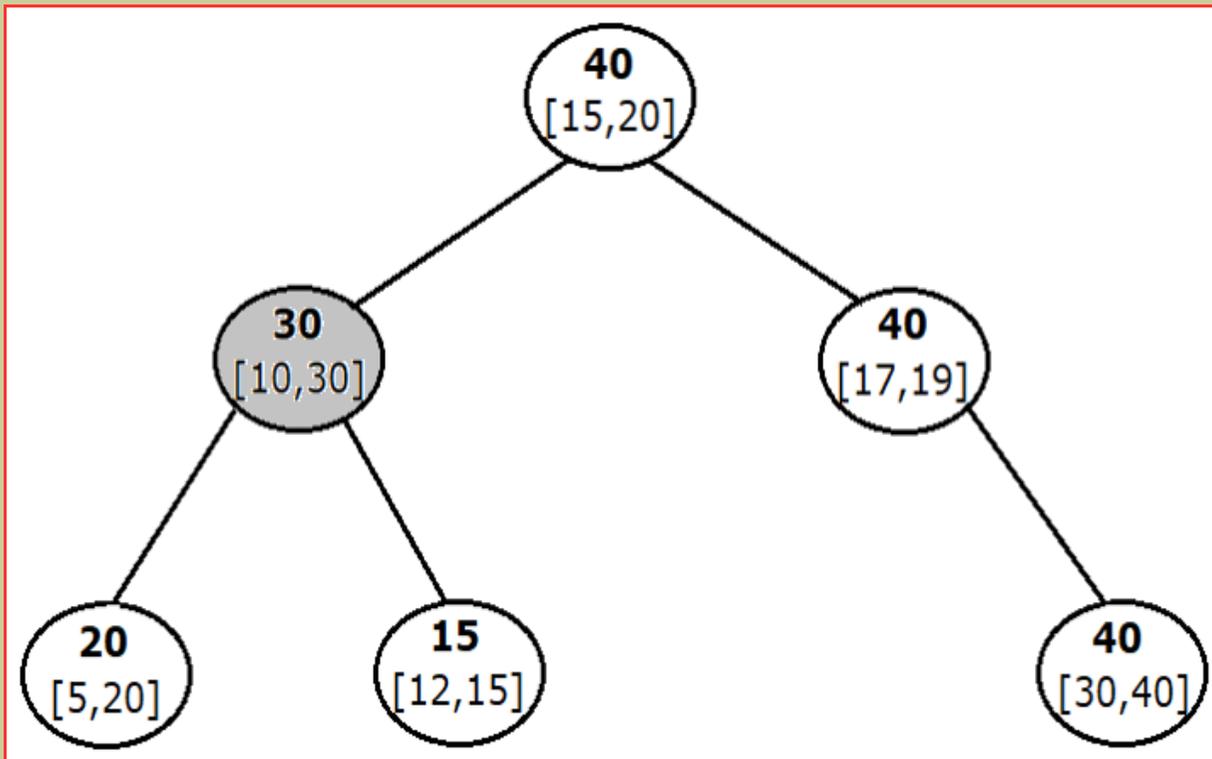
## Remover um intervalo numa árvore de intervalos

- Exemplo: remover o nodo cujo intervalo intercepta com  $[9, 11]$  na árvore em baixo



## Remover um intervalo numa árvore de intervalos

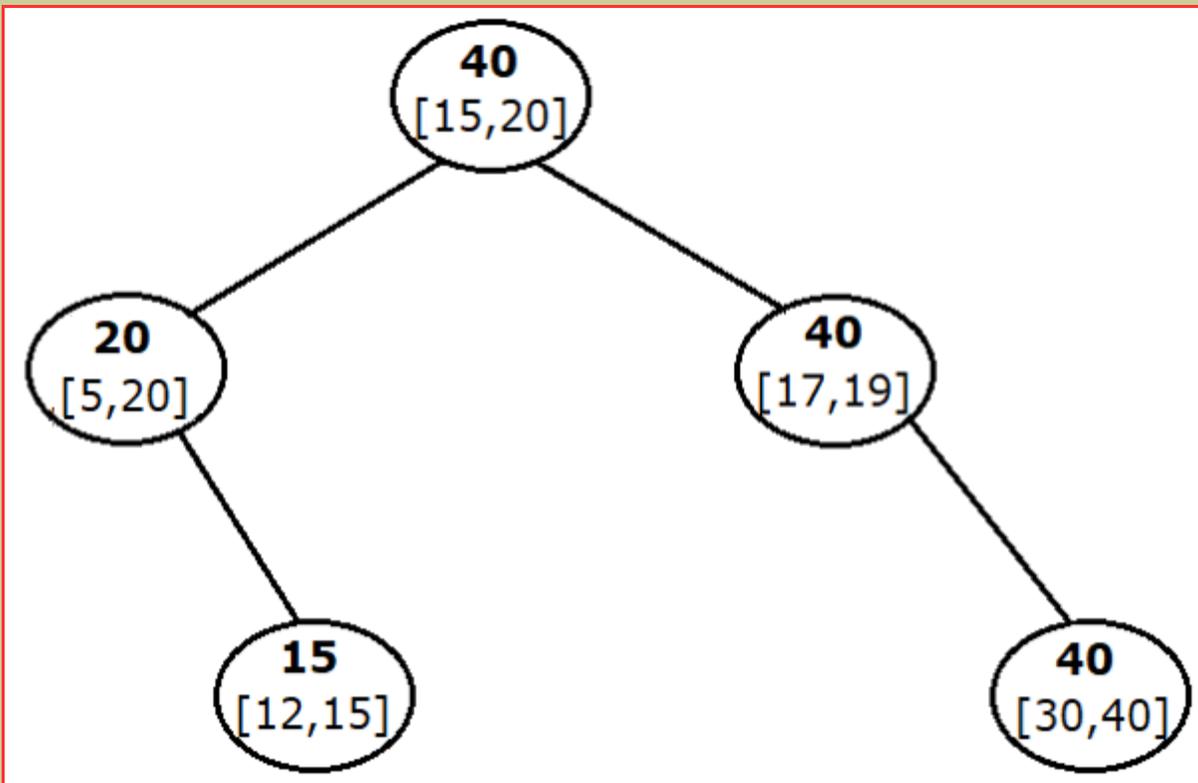
- Exemplo: remover o nodo cujo intervalo intercepta com  $[9, 11]$  na árvore
  - pesquisar pelo nodo da árvore cujo intervalo se sobrepõe ao intervalo a remover:  $T[10,30]$



- o nodo encontrado tem dois filhos, então substituir o nodo pelo descendente
  - mais à direita do seu filho esquerdo (subárvore), ou
  - mais à esquerda do seu filho direito (subárvpre)

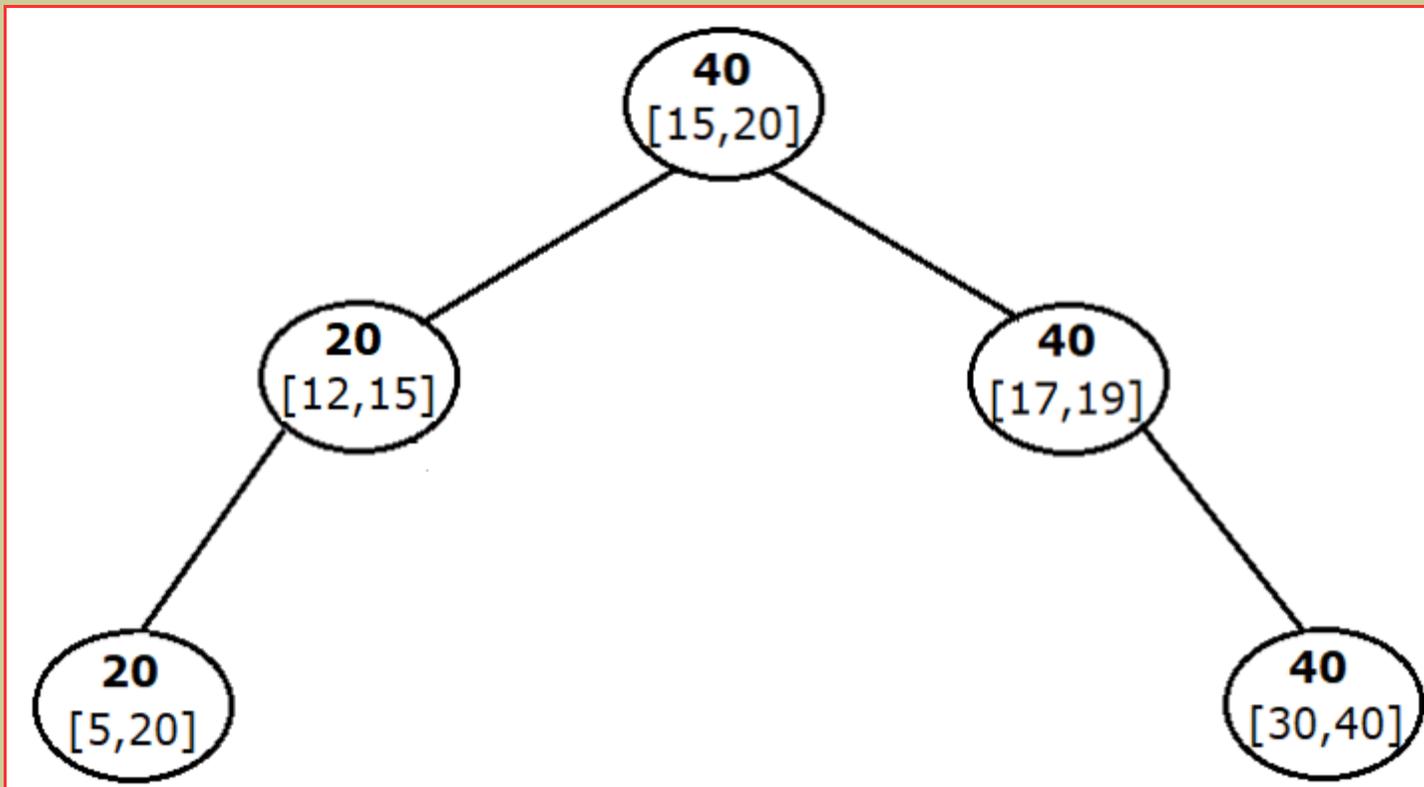
## Remover um intervalo numa árvore de intervalos

- Exemplo: remover o nodo cujo intervalo intercepta com  $[9, 11]$  na árvore
  - substituir o nodo pelo descendente mais à direita do seu filho esquerdo
  - atualizar os nodos antecessores: sem alteração



## Remover um intervalo numa árvore de intervalos

- Exemplo: remover o nodo cujo intervalo intercepta com  $[9, 11]$  na árvore
  - substituir o nodo pelo descendente mais à esquerda do seu filho direito
  - atualizar os nodos antecessores:  $\max(T[12,15]) = \max\{20, 15\} = \mathbf{20}$



## Estrutura de dados aumentadas

### Introdução

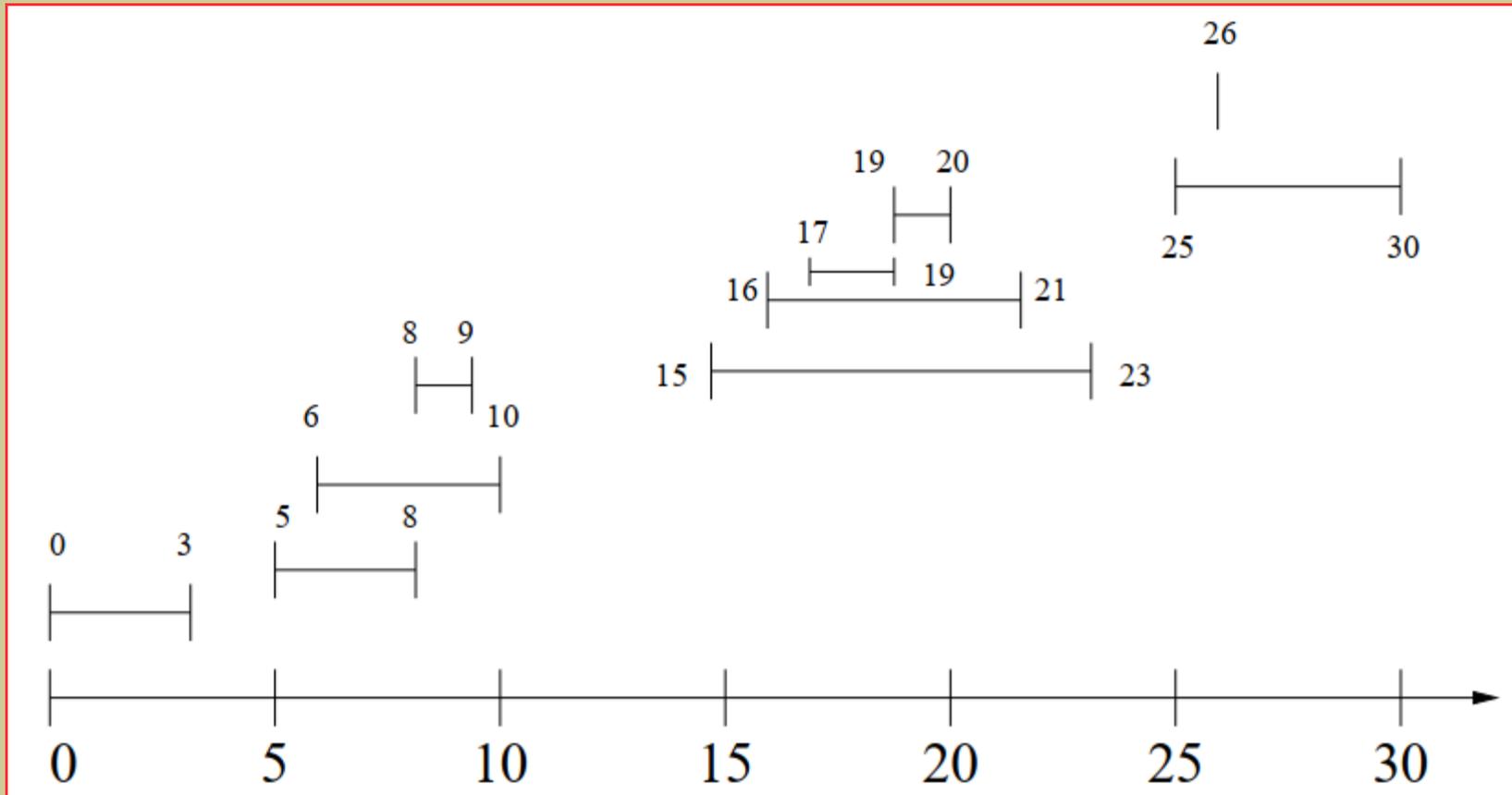
- Dependendo do problema específico a ser resolvido, pode-se obter uma solução mais eficiente e elegante ao optar-se por aumentar uma estrutura já existente, em vez de simplesmente usar uma estrutura já conhecida ou partir-se para a criação de uma totalmente nova
- Os campos a serem acrescentados à estrutura devem ser tais que as suas atualizações sejam bastante eficientes
- Os algoritmos que manipulam a estrutura original, devem agora ser adaptados para terem em consideração a atualização dos novos campos
- Geralmente, os novos campos levam à criação de novos algoritmos que estendam o comportamento da estrutura original para a solução do problema específico

## **Passos envolvidos para aumentar uma estrutura de dados**

- Escolha da estrutura de dados subjacente
- Determinar informação adicional a ser mantida
- Verificar se manutenção da informação adicional pode ser feita pelas operações de modificação da estrutura de dados
- Desenvolver novas operações

## Exemplo

- Aumentar uma árvore preto-vermelho para manipular conjunto de intervalos
- Dado um conjunto de intervalos (em baixo) definir uma representação em forma de árvore



## Exemplo

- Aumentar uma árvore preto-vermelho para manipular conjunto de intervalos
- Dado um conjunto de intervalos, definir uma representação em forma de árvore

