

Algoritmer og Datastrukturer

Dynamisk Rang & Interval Træer [CLRS, kapitel 14]

Dynamisk Rang

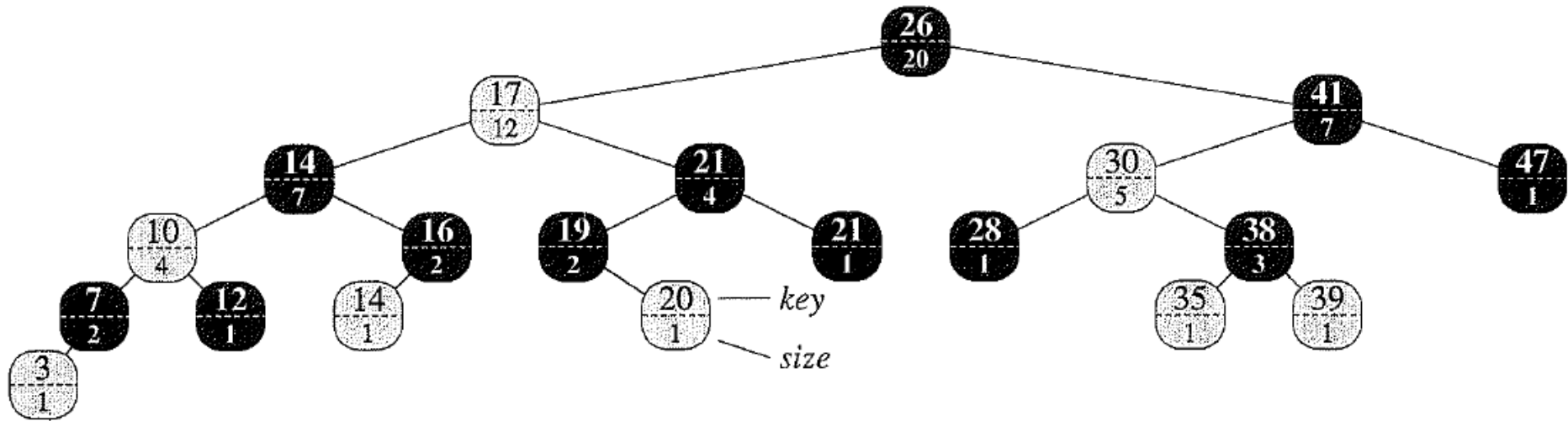
Select(S, i)	$O(\log n)$
Rank(S, x)	
Insert(S, x)	
Delete(S, x)	

Select($S, 9$) = 19

Rank($S, 26$) = 13

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
S	3	7	10	12	13	14	16	17	19	20	21	23	26	28	30	35	38	39	41	47

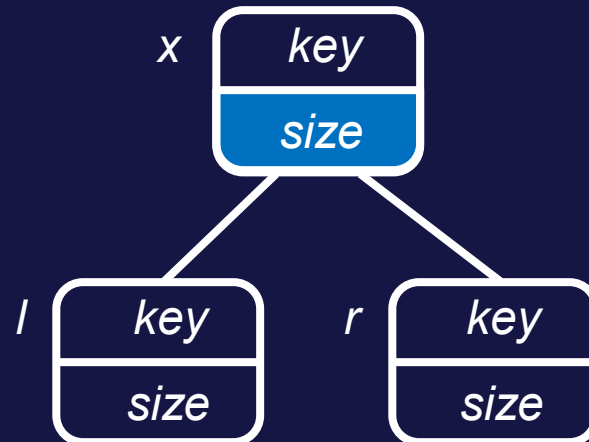
Dynamisk Rang



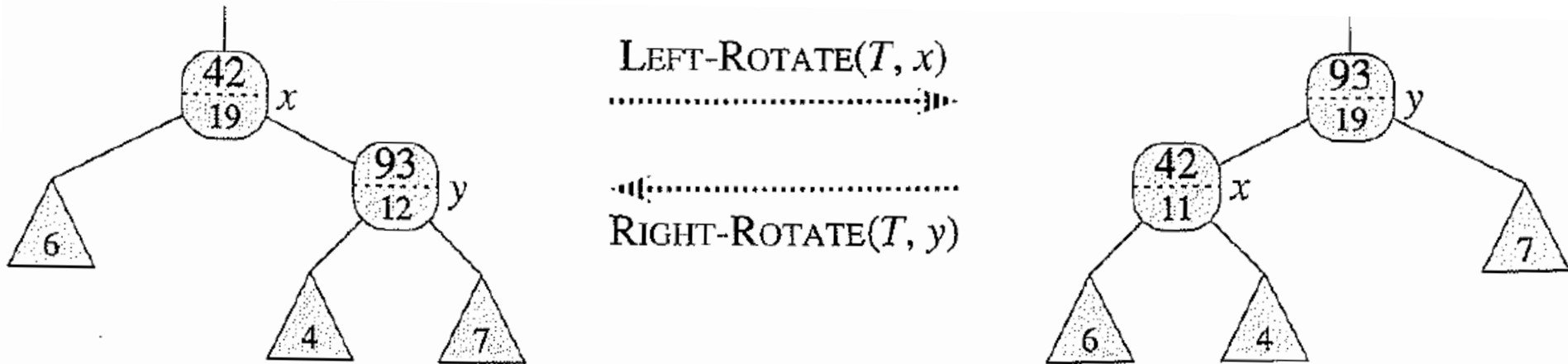
- Find det *i*'te mindste, indsættelser, slettelser
- Vedligehold i rød-sort søgetræ
- Udvid hver knude med *størrelse af undertræerne*

Beregning af Udvidet Information $x.size$?

- a) $x.size = l.size + r.size$
- b) $x.size = l.size + r.size + 1$
- c) $x.size = x.size + l.size + r.size$
- d) $x.size = x.size + l.size + r.size + 1$
- e) ved ikke

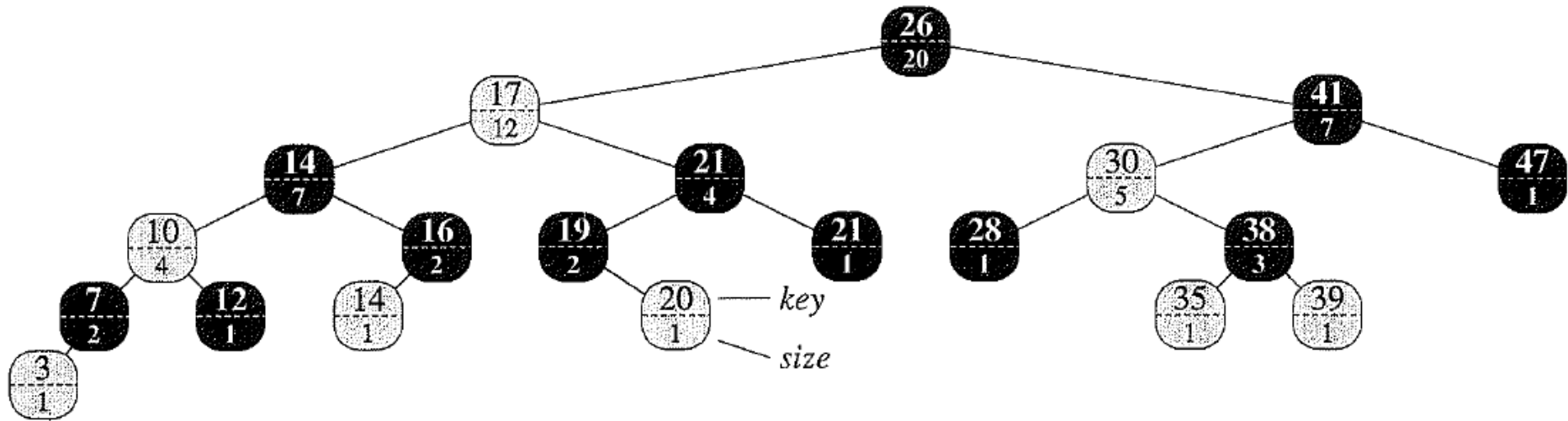


Dynamisk Rang



- Indsættelse/slettelse: opdater **size** på stien til roden
- Under rebalancering af det rød-sortede træ, vedligehold **size** under **rotationer**

Dynamisk Rang



OS-RANK(T, x)

```

1   $r = x.left.size + 1$ 
2   $y = x$ 
3  while  $y \neq T.root$ 
4      if  $y == y.p.right$ 
5           $r = r + y.p.left.size + 1$ 
6       $y = y.p$ 
7  return  $r$ 
    
```

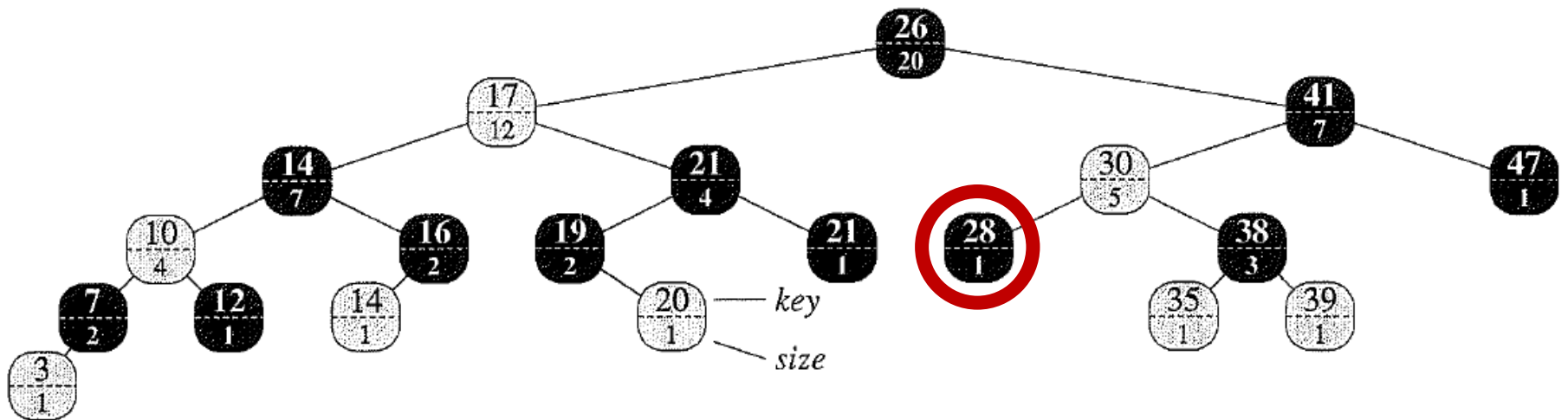
OS-SELECT(x, i)

```

1   $r = x.left.size + 1$ 
2  if  $i == r$ 
3      return  $x$ 
4  elseif  $i < r$ 
5      return OS-SELECT( $x.left, i$ )
6  else return OS-SELECT( $x.right, i - r$ )
    
```

Rang af 28 ?

- a) 13
- b) 14
- c) 16
- d) 28
- e) ved ikke

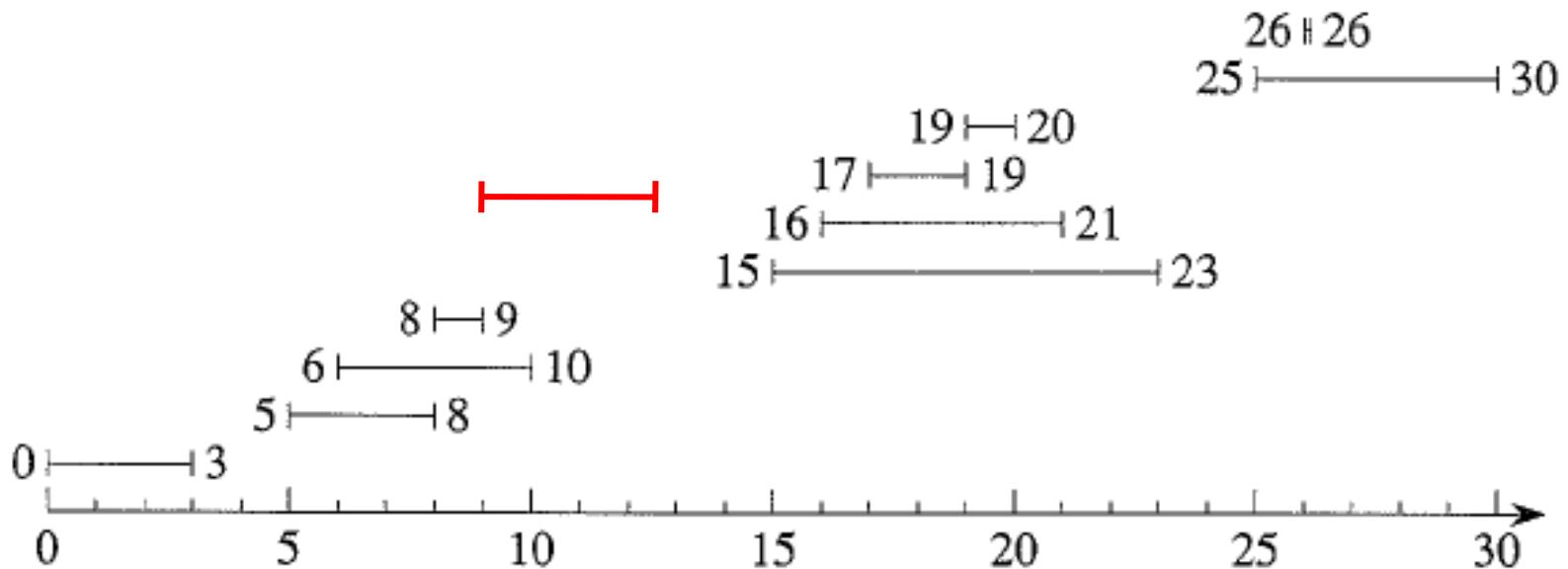


Dynamisk Rang

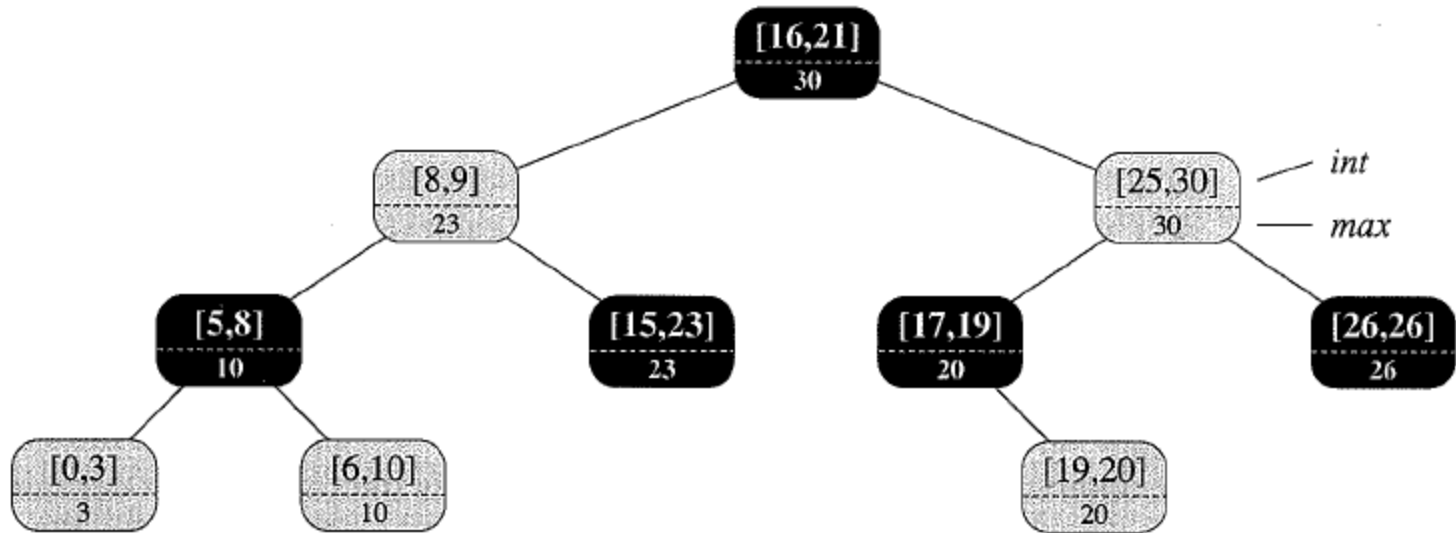
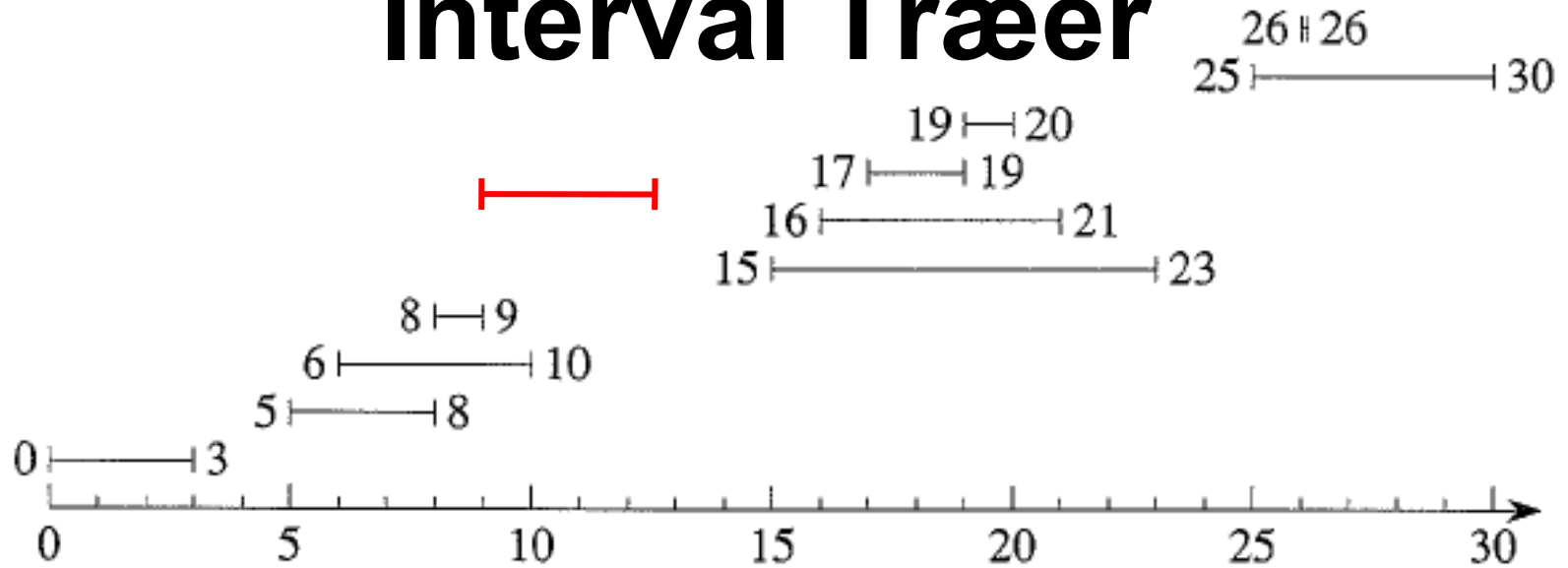
Select(S, i)	$O(\log n)$
Rank(S, x)	
Insert(S, x)	
Delete(S, x)	

Interval Træer

- Vedligehold en mængde af intervaller
- Indsæt og slet indsatte intervaller
- Find et interval der overlapper med et givet **interval**



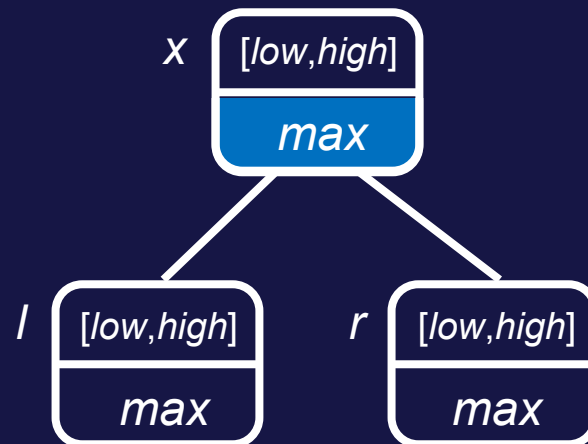
Interval Træer



- Søgetræ over intervallernes **venstre endepunkt**
- Hver knude gemmer yderligere **maximum højre endepunkt** for et interval i undertræet

Beregning af Udvidet Information $x.max$?

- a) $x.max = r.max$
- b) $x.max = r.high$
- c) $x.max = \max(r.max, x.high)$
- d) $x.max = \max(l.max, x.high)$
- e) $x.max = \max(l.max, r.max, x.high)$
- f) ved ikke

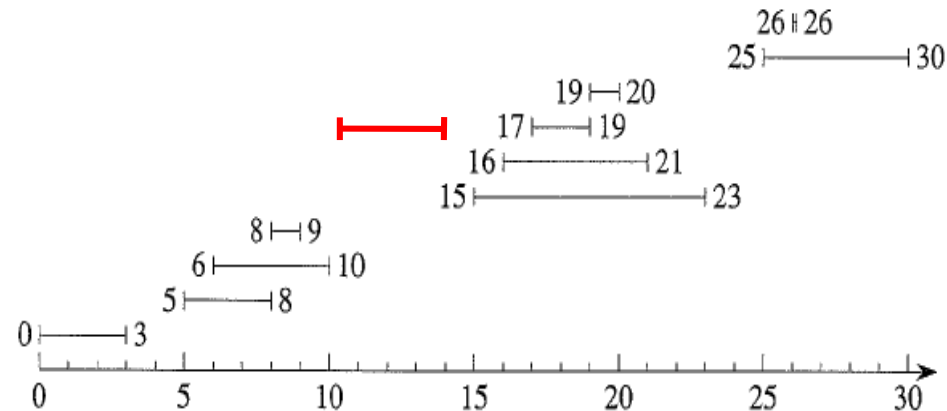
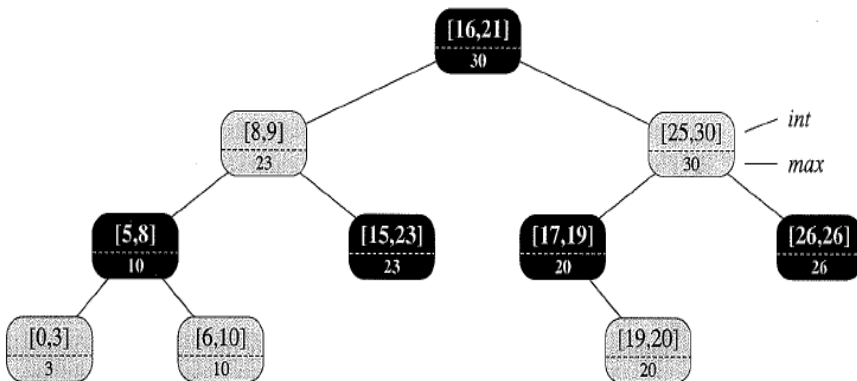


Interval Træer

INTERVAL-SEARCH(T, i)

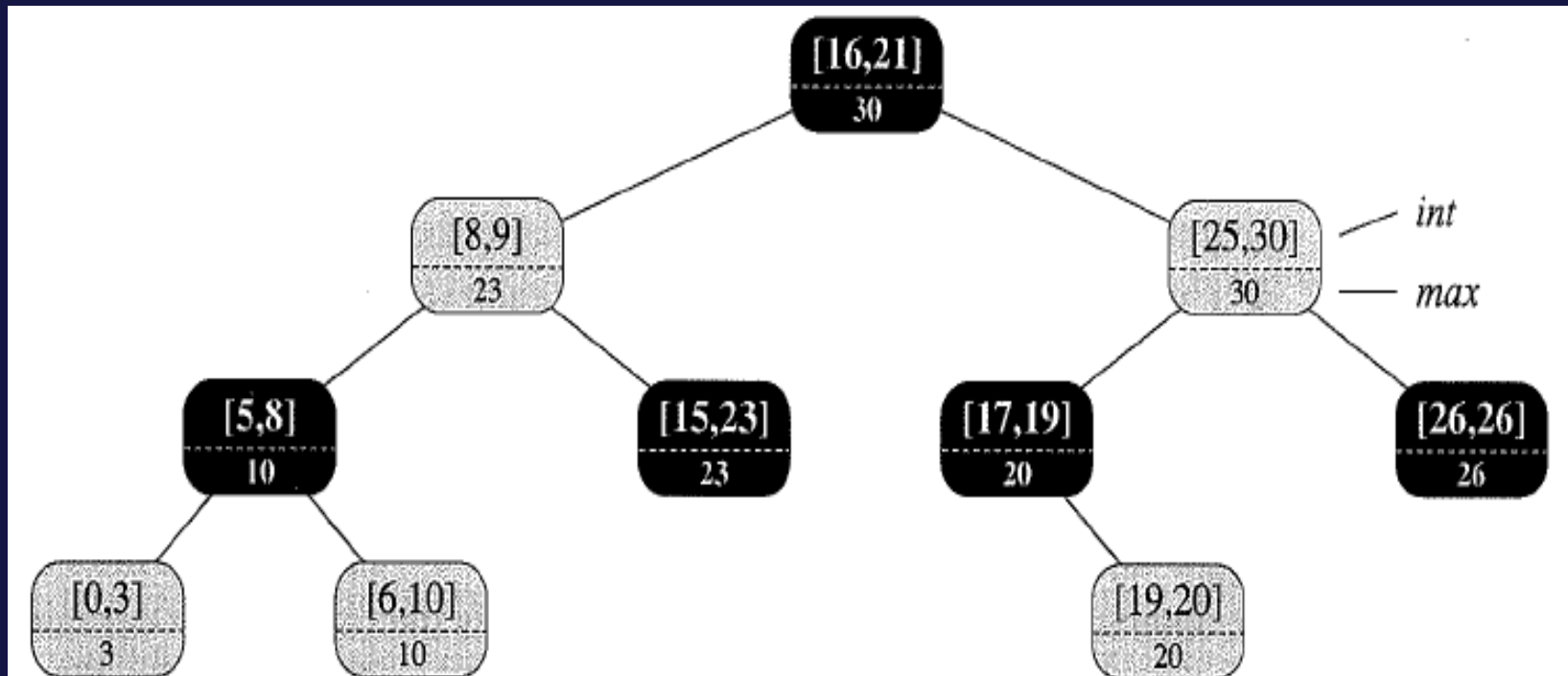
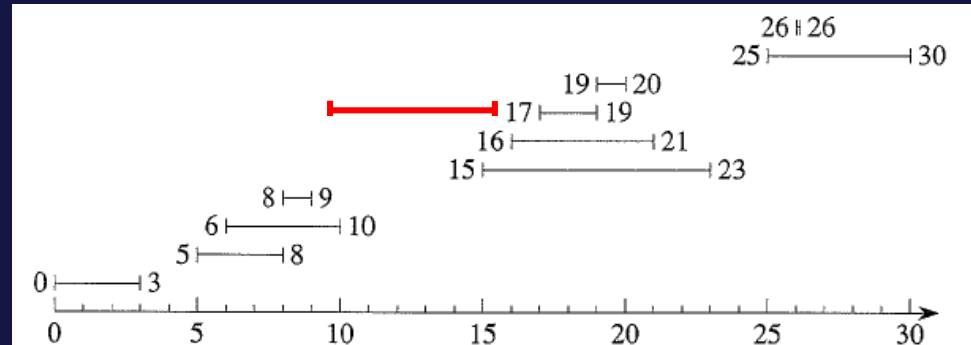
```
1  $x = T.root$ 
2 while  $x \neq T.nil$  and  $i$  does not overlap  $x.int$ 
3   if  $x.left \neq T.nil$  and  $x.left.max \geq i.low$ 
4      $x = x.left$ 
5   else  $x = x.right$ 
6 return  $x$ 
```

tricky



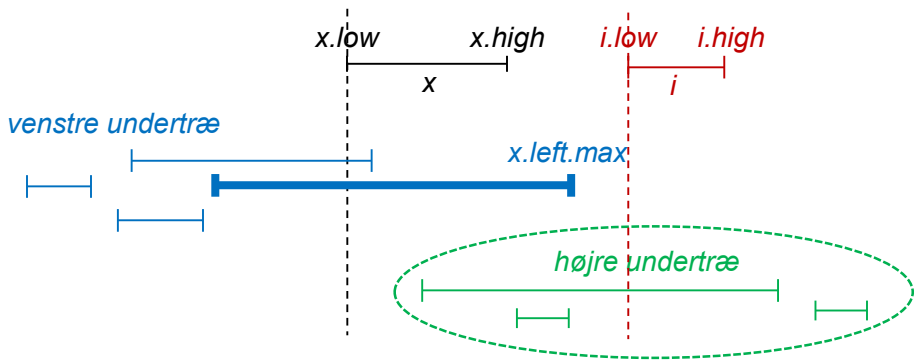
INTERVAL-SEARCH($T, [9.5, 15.5]$) ?

- a) [6, 10]
- b) [15, 23]
- c) ved ikke



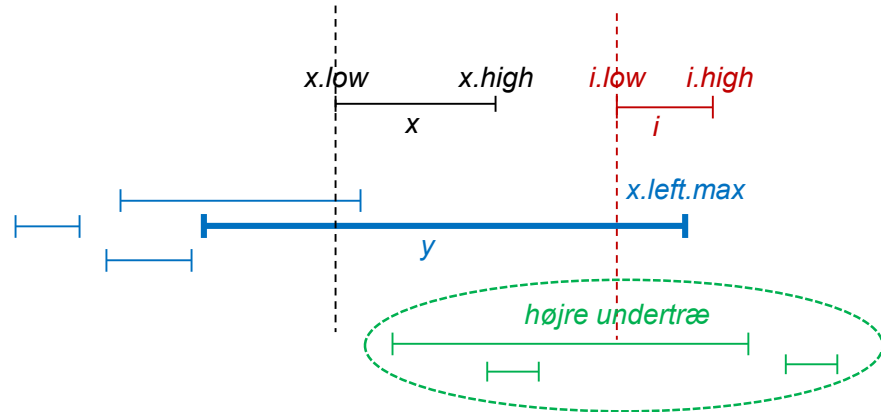
Interval Træer - Korrekthed

x til venstre for i og $x.left.max < i.low$



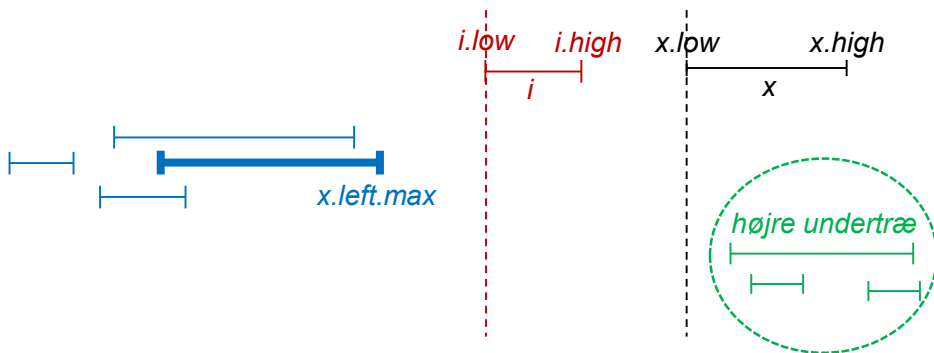
Kan ikke være overlap i venstre undertræ, så sikkert at **gå til højre**

x til venstre for i og $x.left.max \geq i.low$



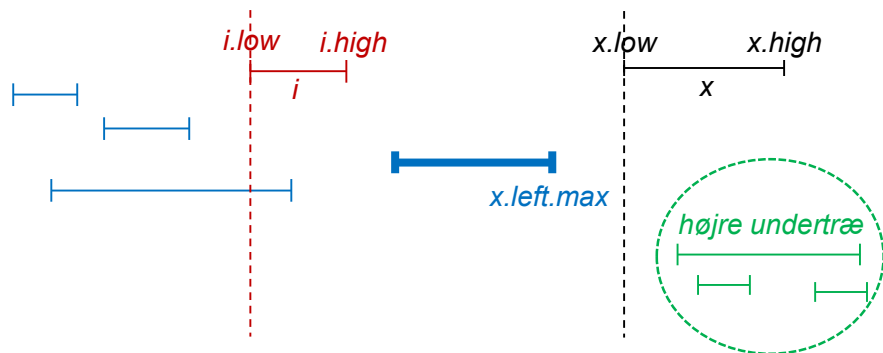
Findes overlappende interval y i venstre undertræ, så sikkert at **gå til venstre**

i til venstre for x og $x.left.max < i.low$



Ingen overlap (overser ikke noget ved at **gå til højre**)

i til venstre for x og $x.left.max \geq i.low$



Ingen overlap i højre undertræ, så sikkert at **gå til venstre**

Interval Træer

Search(T, i)	$O(\log n)$
Insert(T, i)	
Delete(T, i)	