



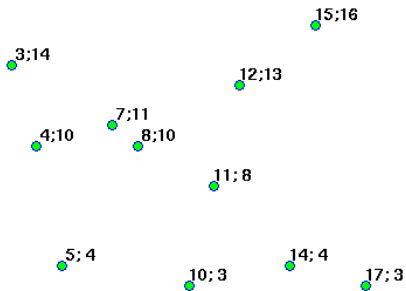
**OI-OPPA. European Social Fund  
Prague & EU: We invest in your future.**

---

# 2D range tree construction and range tree search

Martin Milichovský

FEL ČVUT



```

procedure buildTree(pts, level, n)
  if size(pts)==1
    return pts[0]
  newNode := new TreeNode()
  dim := level%n
  pivot := median value of pts in dim
  newNode.label = pivot
  newNode.left := ...
    buildTree({p ∈ pts | p[dim] ≤ pivot}, level+1, n)
  newNode.right := ...
    buildTree({p ∈ pts | p[dim] > pivot}, level+1, n)
  return newNode
end procedure

```

3;14

4;10

5;4

7;11

8;10

10;3

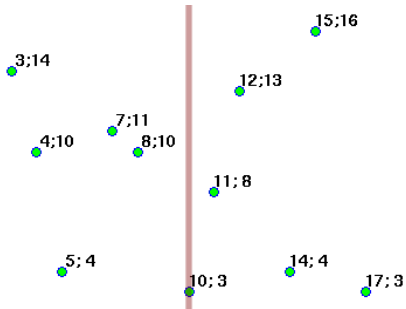
11;8

12;13

14;4

15;16

17;3



```

procedure buildTree(pts, level, n)
  if size(pts)==1
    return pts[0]
  newNode := new TreeNode()
  dim := level%n
  pivot := median value of pts in dim
  newNode.label = pivot
  newNode.left := ...
    buildTree({p ∈ pts | p[dim] ≤ pivot}, level+1, n)
  newNode.right := ...
    buildTree({p ∈ pts | p[dim] > pivot}, level+1, n)
  return newNode
end procedure

```

3;14

4;10

5;4

7;11

8;10

10;3

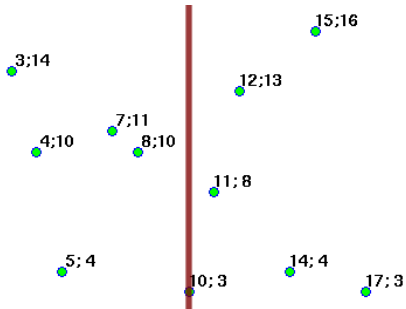
11;8

12;13

14;4

15;16

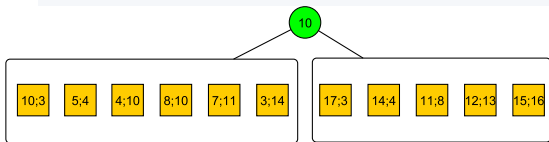
17;3

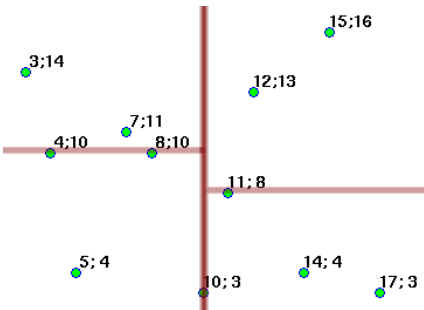


```

procedure buildTree(pts, level, n)
  if size(pts)==1
    return pts[0]
  newNode := new TreeNode()
  dim := level%n
  pivot := median value of pts in dim
  newNode.label = pivot
  newNode.left := ...
    buildTree({p ∈ pts | p[dim] ≤ pivot}, level+1, n)
  newNode.right := ...
    buildTree({p ∈ pts | p[dim] > pivot}, level+1, n)
  return newNode
end procedure

```

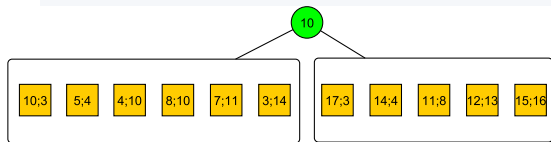


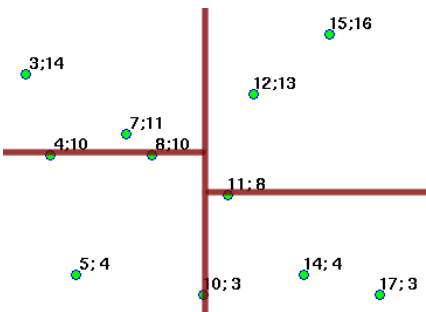


```

procedure buildTree(pts, level, n)
  if size(pts)==1
    return pts[0]
  newNode := new TreeNode()
  dim := level%n
  pivot := median value of pts in dim
  newNode.label = pivot
  newNode.left := ...
    buildTree({p ∈ pts | p[dim] ≤ pivot}, level+1, n)
  newNode.right := ...
    buildTree({p ∈ pts | p[dim] > pivot}, level+1, n)
  return newNode
end procedure

```

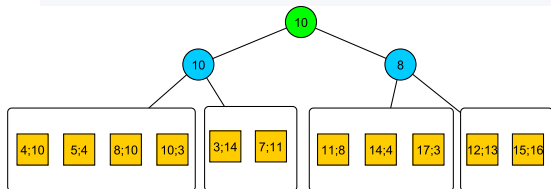


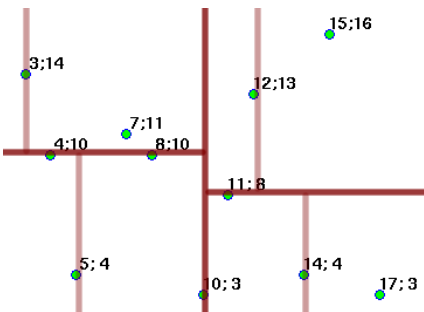


```

procedure buildTree(pts, level, n)
  if size(pts)==1
    return pts[0]
  newNode := new TreeNode()
  dim := level%n
  pivot := median value of pts in dim
  newNode.label = pivot
  newNode.left := ...
    buildTree({p ∈ pts|p[dim] ≤ pivot},level+1,n)
  newNode.right := ...
    buildTree({p ∈ pts|p[dim] > pivot},level+1,n)
  return newNode
end procedure

```

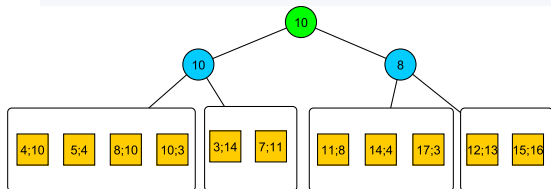




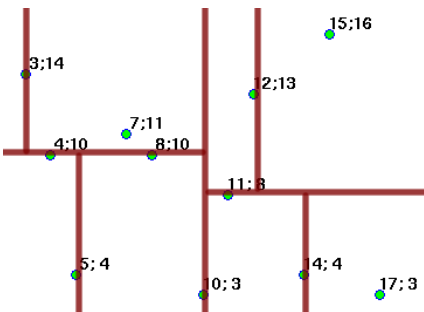
```

procedure buildTree(pts, level, n)
  if size(pts)==1
    return pts[0]
  newNode := new TreeNode()
  dim := level%n
  pivot := median value of pts in dim
  newNode.label = pivot
  newNode.left := ...
    buildTree({p ∈ pts | p[dim] ≤ pivot}, level+1, n)
  newNode.right := ...
    buildTree({p ∈ pts | p[dim] > pivot}, level+1, n)
  return newNode
end procedure

```



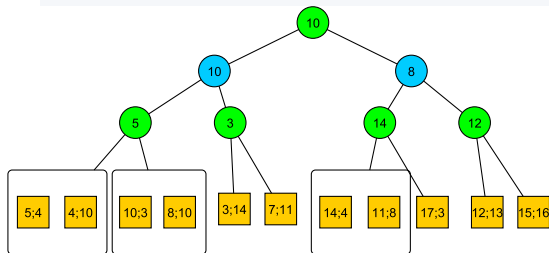


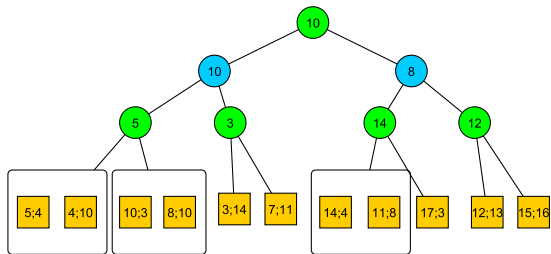
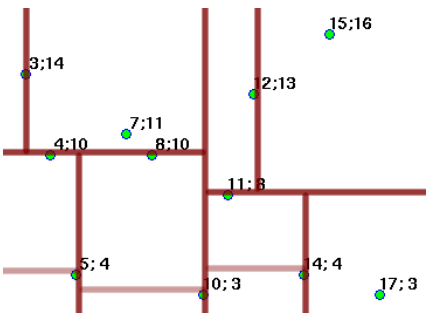


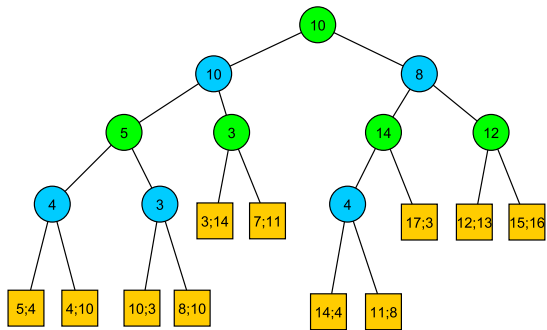
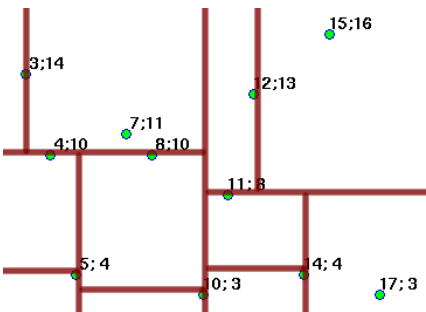
```

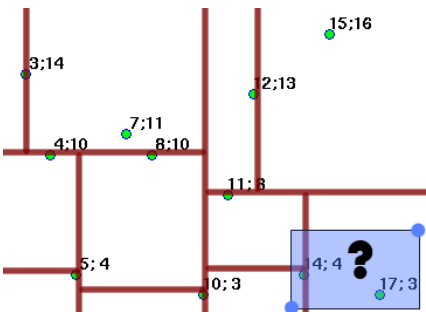
procedure buildTree(pts, level, n)
  if size(pts)==1
    return pts[0]
  newNode := new TreeNode()
  dim := level%n
  pivot := median value of pts in dim
  newNode.label = pivot
  newNode.left := ...
    buildTree({p ∈ pts|p[dim] ≤ pivot},level+1,n)
  newNode.right := ...
    buildTree({p ∈ pts|p[dim] > pivot},level+1,n)
  return newNode
end procedure

```







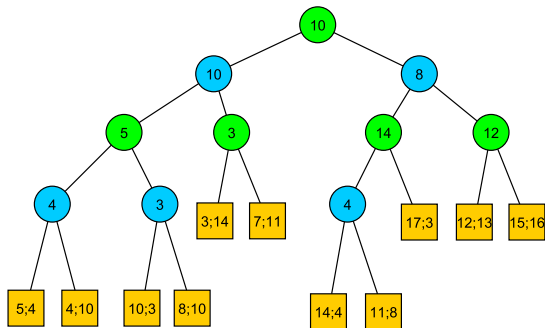


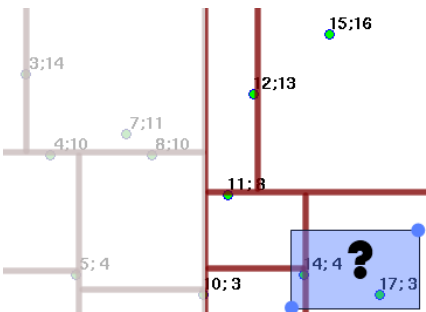
## Range query

e.g. find all nodes having

- $13 \leq x \leq 19$  and
- $2 \leq y \leq 6$ .

Search the tree and skip subtrees lying out of desired interval.



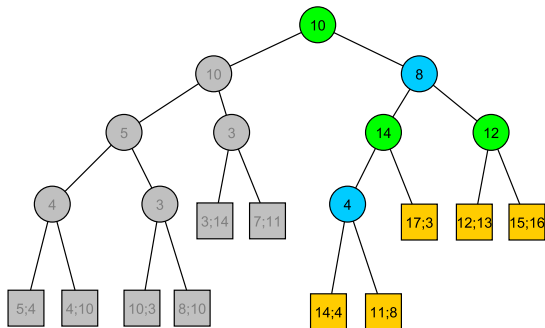


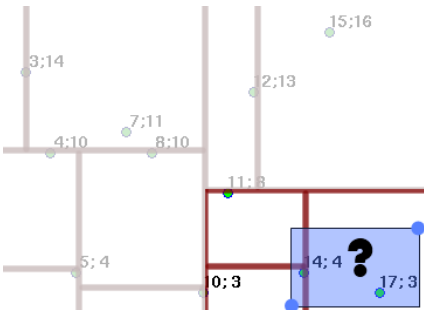
## Range query

e.g. find all nodes having

- $13 \leq x \leq 19$  and
- $2 \leq y \leq 6$ .

Search the tree and skip subtrees lying out of desired interval.



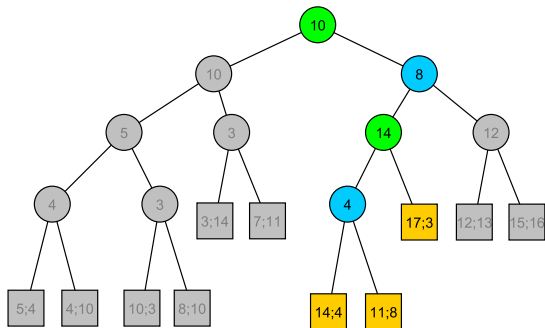


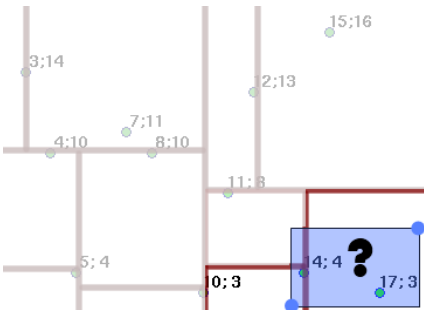
## Range query

e.g. find all nodes having

- $13 \leq x \leq 19$  and
- $2 \leq y \leq 6$ .

Search the tree and skip subtrees lying out of desired interval.



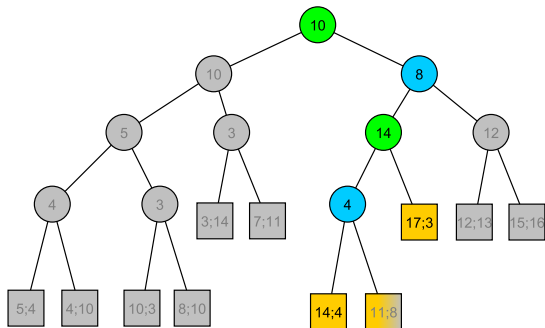


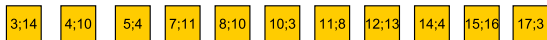
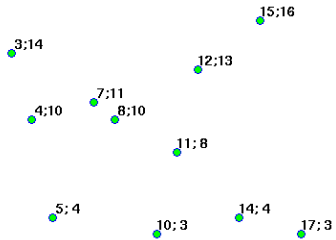
## Range query

e.g. find all nodes having

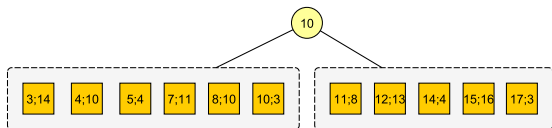
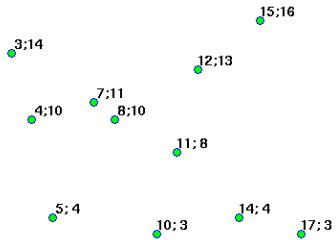
- $13 \leq x \leq 19$  and
- $2 \leq y \leq 6$ .

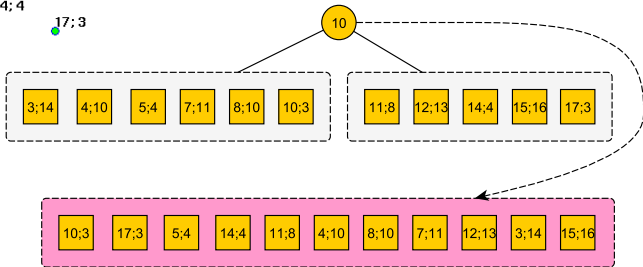
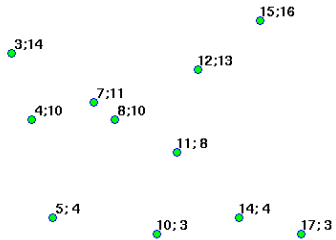
Search the tree and skip subtrees lying out of desired interval.

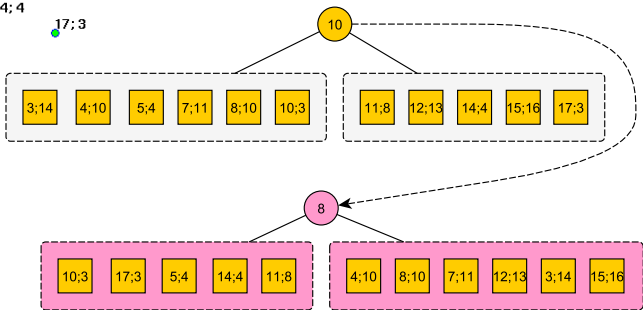
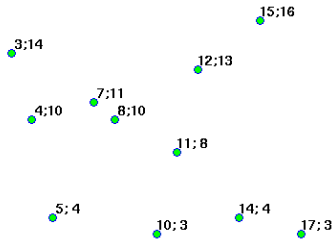


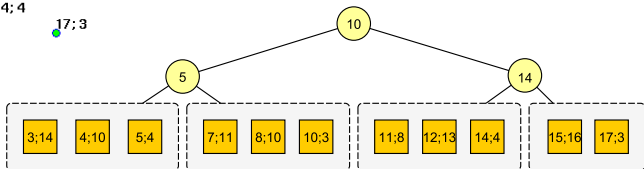
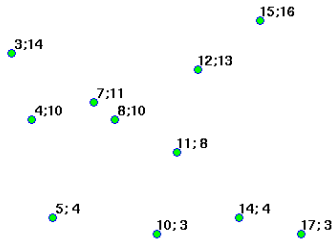


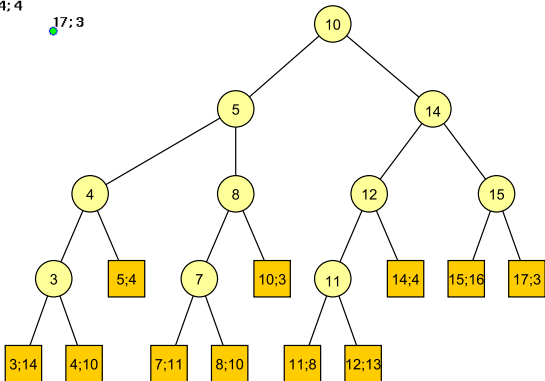
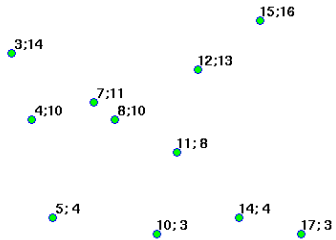


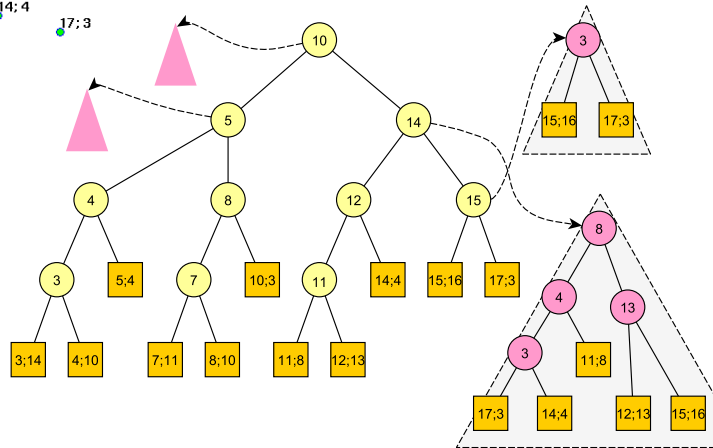
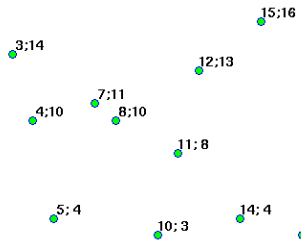


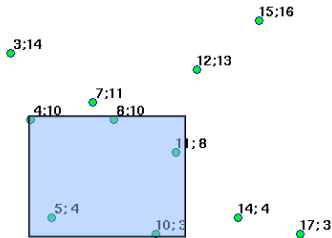










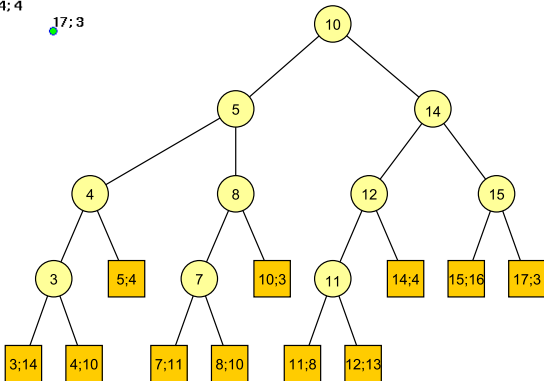


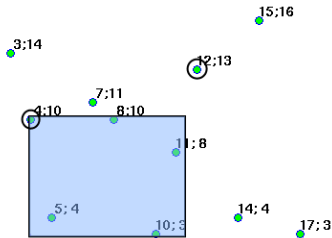
## Range query

e.g. find all nodes having

$$4 \leq x \leq 12 \text{ and}$$

$$3 \leq y \leq 10.$$



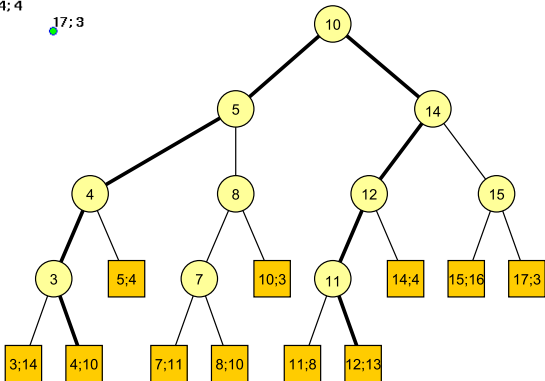


## Range query

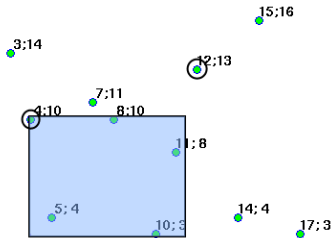
e.g. find all nodes having

$$4 \leq x \leq 12 \text{ and}$$

$$3 \leq y \leq 10.$$





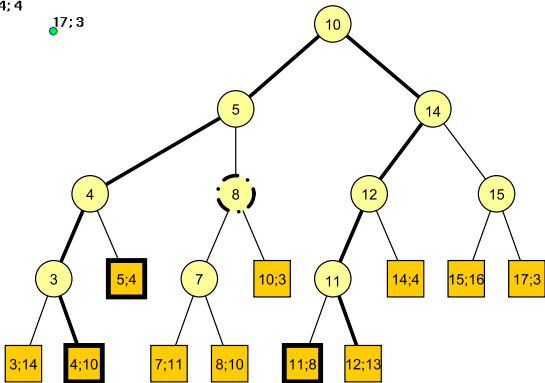


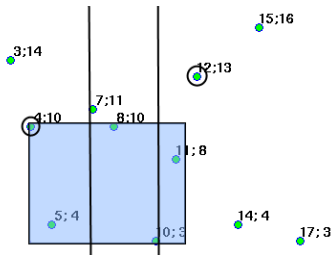
## Range query

e.g. find all nodes having

$$4 \leq x \leq 12 \text{ and}$$

$$3 \leq y \leq 10.$$



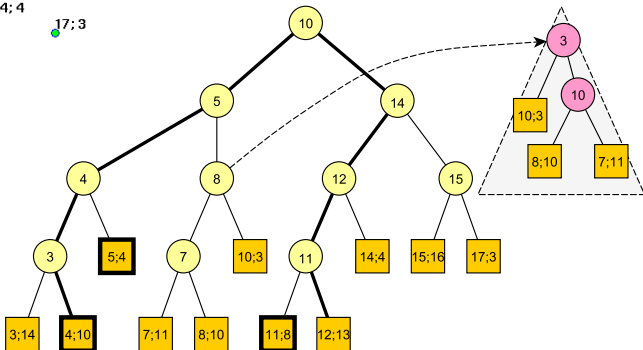


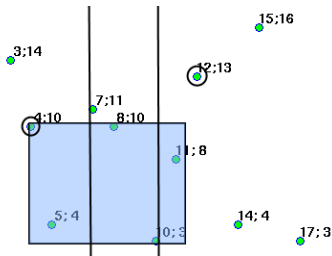
## Range query

e.g. find all nodes having

$$4 \leq x \leq 12 \text{ and}$$

$$3 \leq y \leq 10.$$



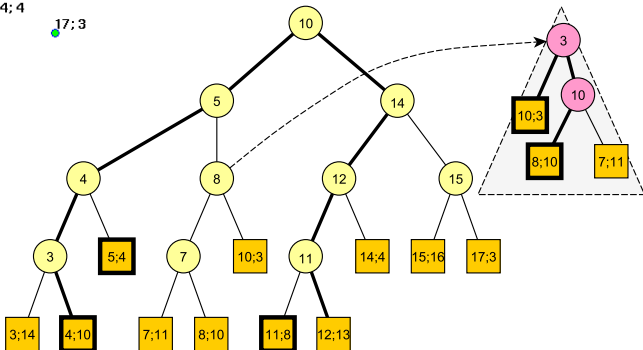


## Range query

e.g. find all nodes having

$$4 \leq x \leq 12 \text{ and}$$

$$3 \leq y \leq 10.$$





**OI-OPPA. European Social Fund  
Prague & EU: We invest in your future.**

---