

# Digitized Ordered + Spatial Collections

Note Title

11/6/2007

## Methods added to Ordered Collection:

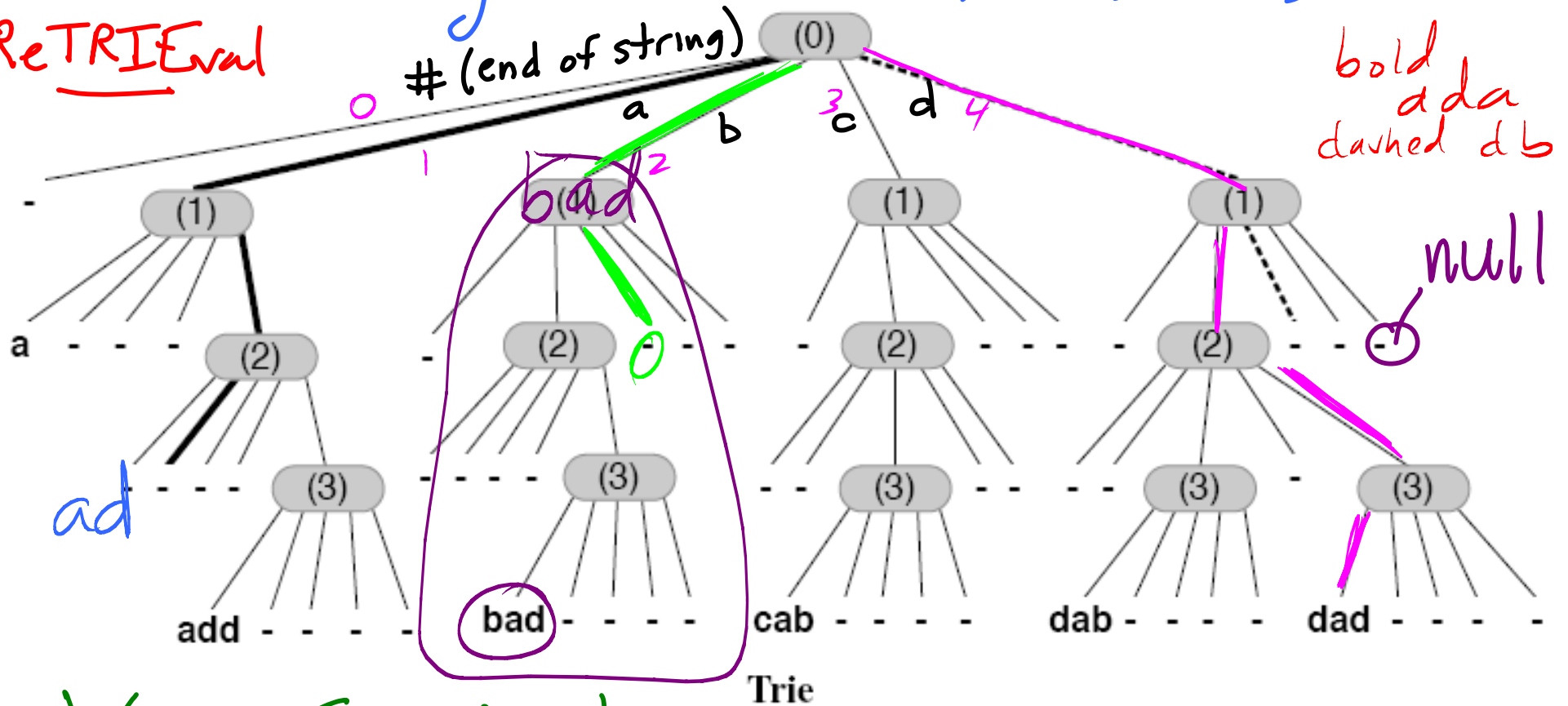
mutates it

**void completions**(*E prefix*, *Collection*<? *super E*> *c*): Appends all elements in this collection that have the given prefix to the given collection *c*. We consider an element to be a prefix of itself.

**void longestCommonPrefix**(*E element*, *Collection*<? *super E*> *c*): Appends to the provided collection *c* all elements in this collection that have a longest common prefix with *element*.

Trie holding  $\langle a\#, add\#, bad\#, cab\#, dab\#, dad\# \rangle$

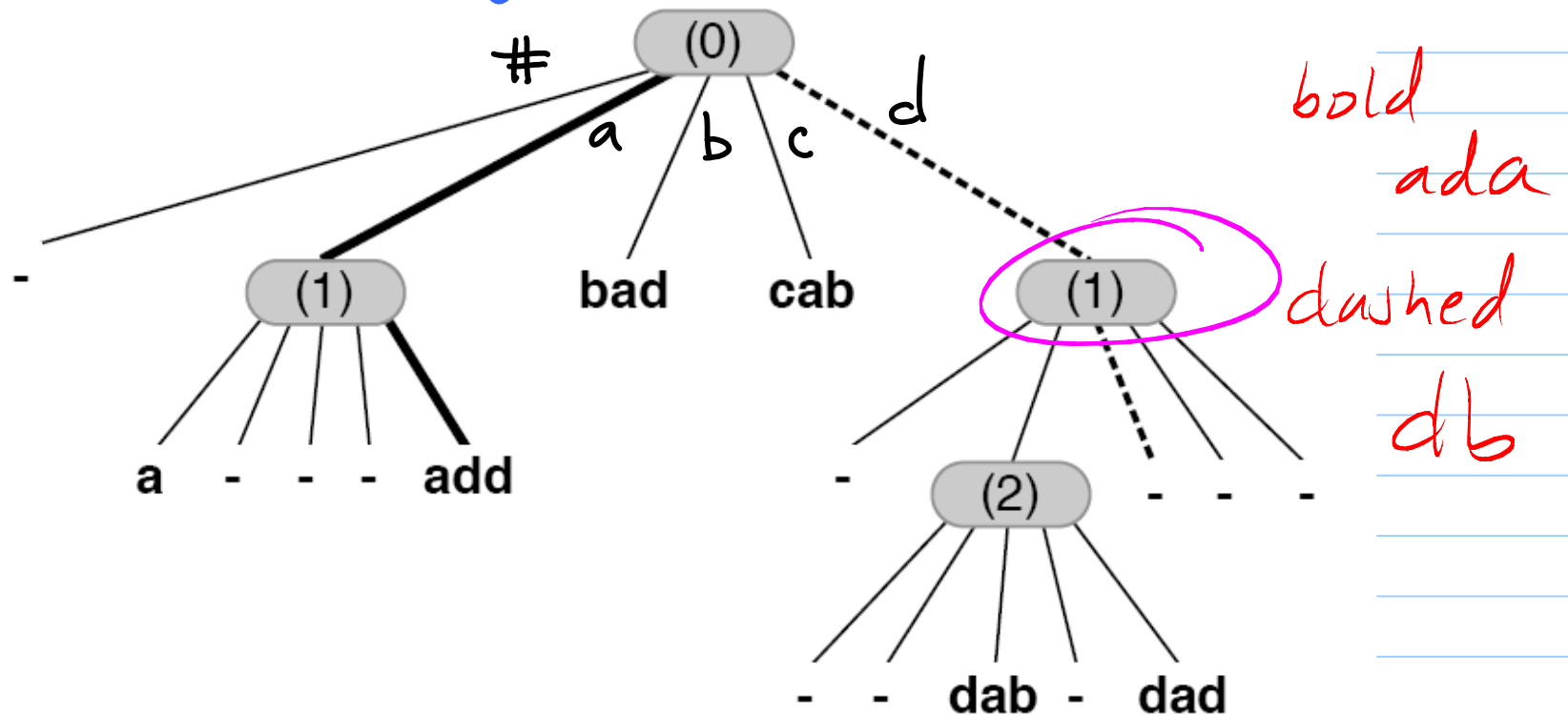
ReTRIEval



Very Simple to program

Drawback - Wastes a lot of space

Compact Trie holding  $\langle a, add, bad, cab, dab, dad \rangle$

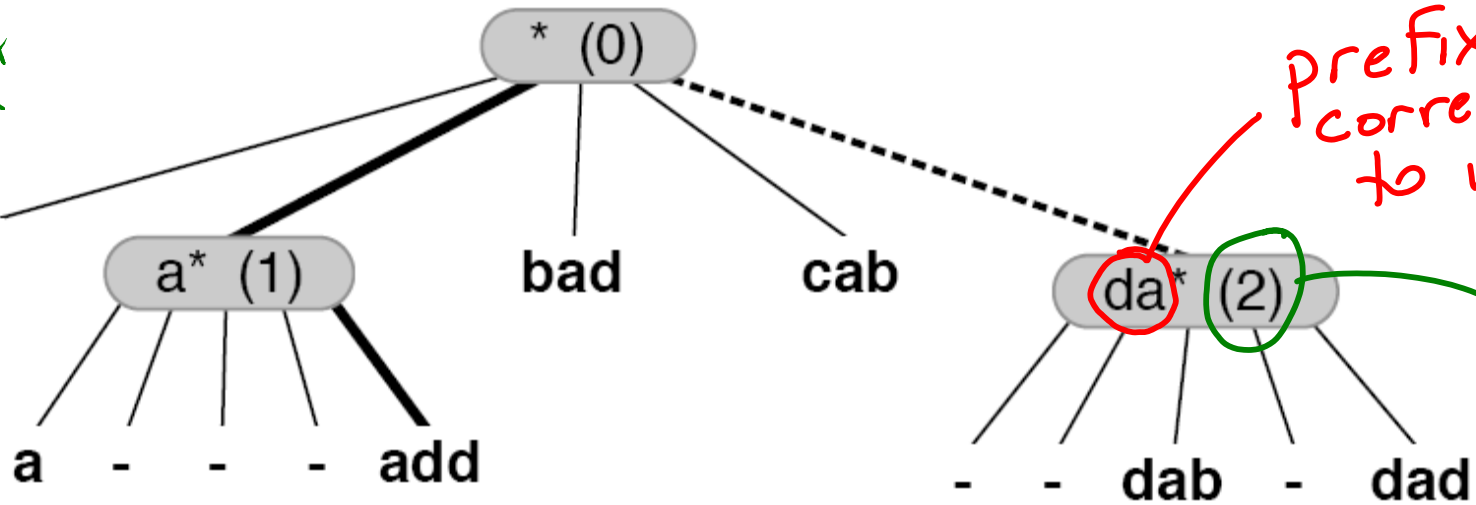


Compact Trie

A little more complex to code, but more space efficient

# Compressed Trie holding $\langle a, add, bad, cab, dab, dad \rangle$

ddd  
↑ ↑ ↑  
0 1 2



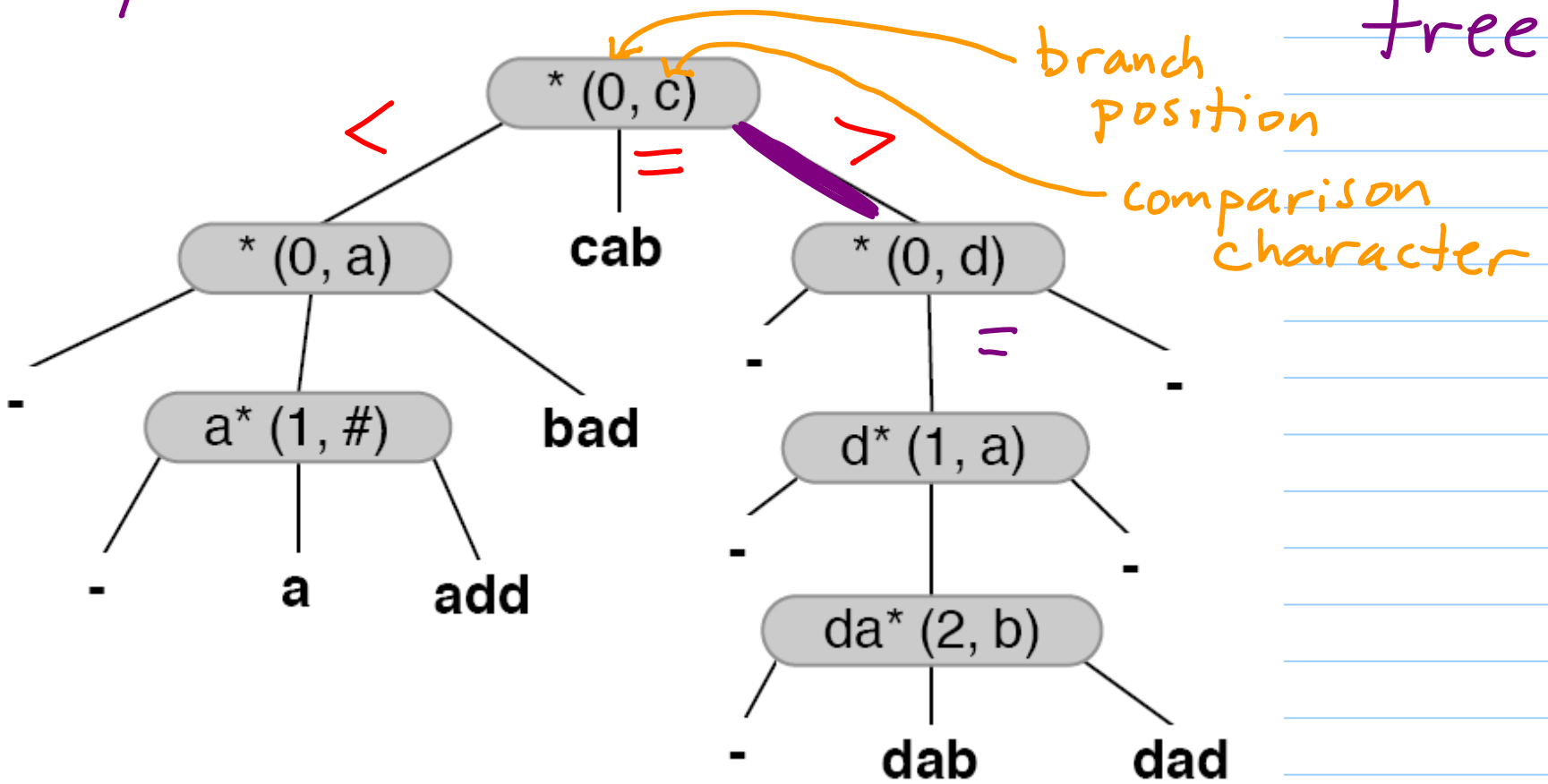
Compressed Trie

Slightly more complex but  
even more space efficient  
(usually)

branch  
position  
(char to use  
for next  
branch)

# TST (Ternary Search Trie)

Hybrid between trie + binary search tree



Suffix Tree - insert all suffixes of a set of words into a Trie

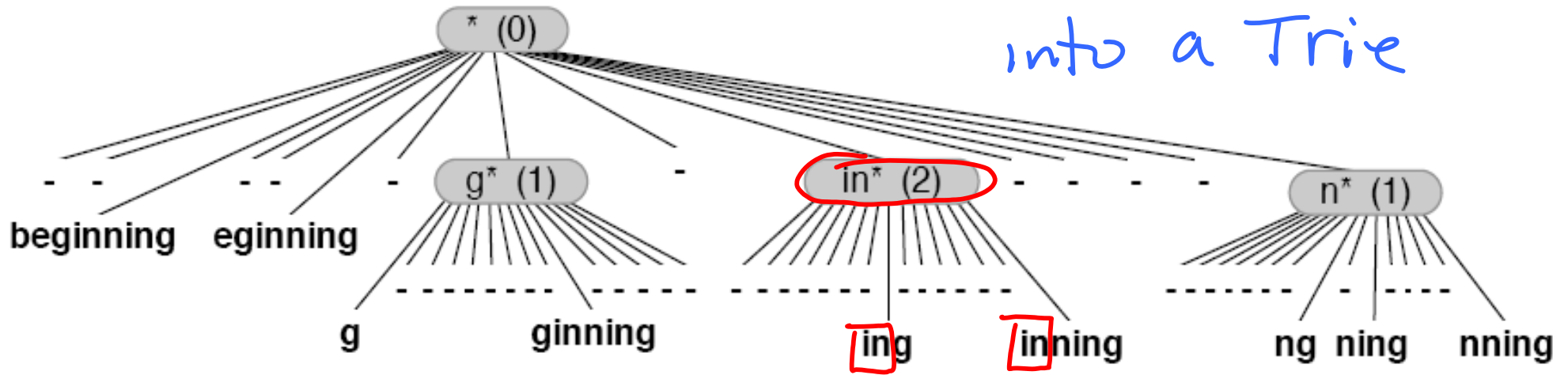


Table 39.8 A suffix tree for the word "beginning" using a compressed trie.

in

# Indexing Trie

Leaves just give a location  
(offset in characters) into  
document

# Spatial Collection - add following methods to Collection Interface

***SpatialCollection(Comparator<? super E> ... comparators)***: Creates a new empty spatial collection, where the provided comparators define the dimensions along which data are compared. Each dimension is assigned an index (0, 1, ...) that is fixed according to the order in which the comparators are provided as parameters to the constructor.

*k dim*

*dim*

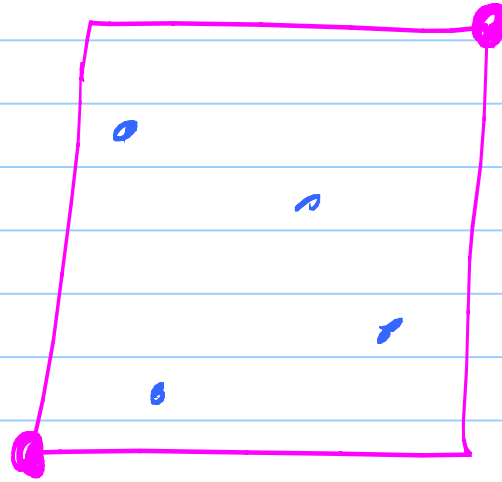
***E max(int dimension)***: Returns a greatest element in the collection along the given dimension. This method throws a *NoSuchElementException* when the collection is empty. It throws an *IllegalArgumentException* when the given dimension index is not valid for this spatial collection.

*0, 1, ..., k-1*

***E min(int dimension)***: Returns a least element in the collection along the given dimension. This method throws a *NoSuchElementException* when the collection is empty. It throws an *IllegalArgumentException* when the given dimension index is not valid for this spatial collection.

***Collection<E> withinBounds(E minCorner, E maxCorner)***: Returns a collection of the elements that fall within (or on) the boundary of the multidimensional box defined by the two given corners, *minCorner* and *maxCorner*. That is, this method performs an orthogonal range search. It requires that the coordinates of *minCorner* are less than or equal to those of *maxCorner* along every dimension of the spatial collection.

min x

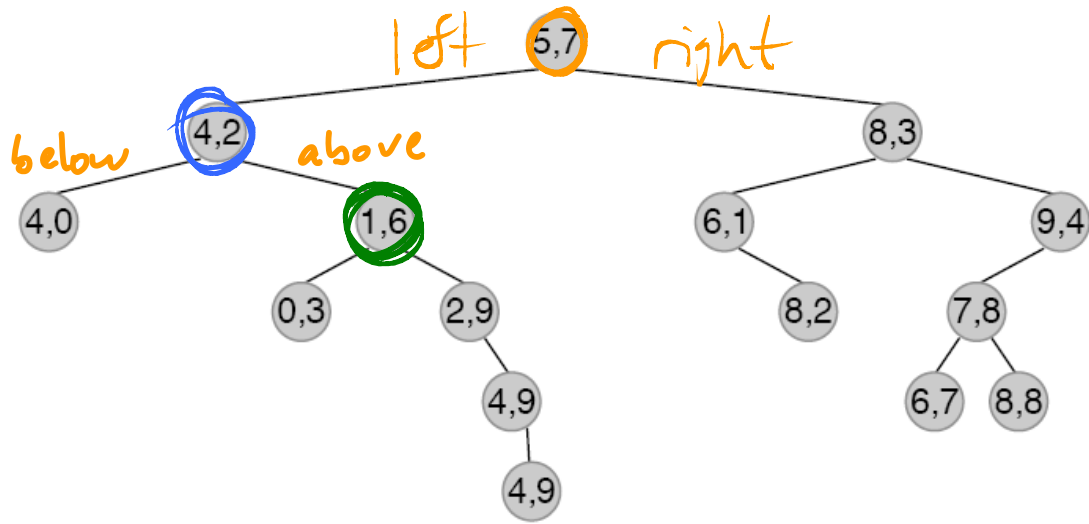


min y

2d  
values  
to  
specify  
box

axis-aligned box

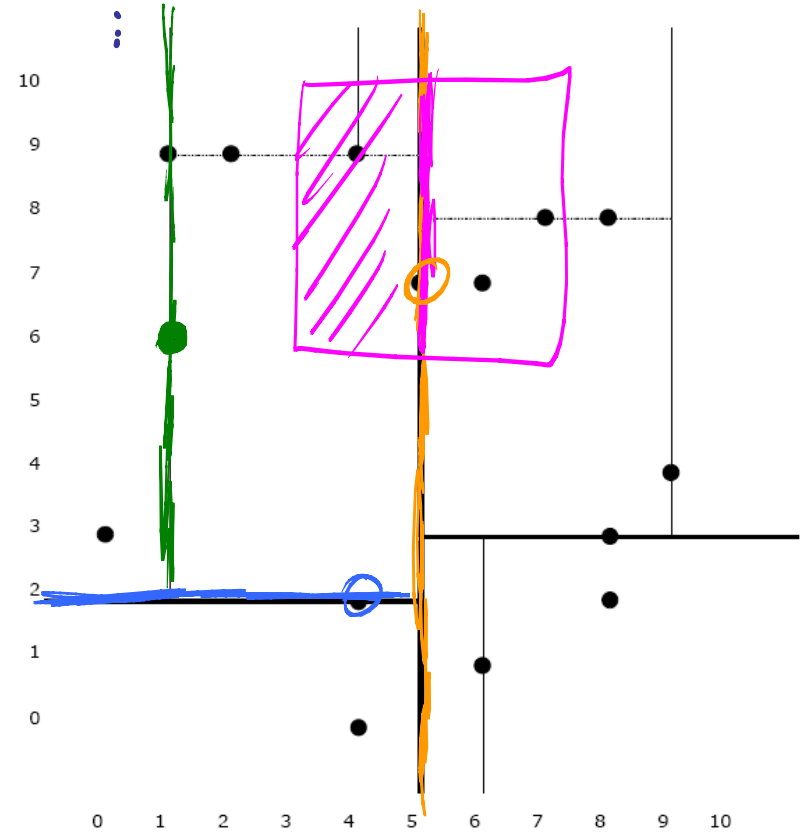
# K-d Tree for $K$ (#dim) = 2



level 0

level 1

level 2



at even level branch  
Using x-coord

at odd level branch  
Using y-coord

Cost of within bounds

$$O\left(n^{d-1/d} + \# \text{ pts in box}\right)$$

$$d=2 \quad O\left(\sqrt{n} + \# \text{ pts}\right)$$

$$d=3 \quad O\left(n^{2/3} + \# \text{ pts}\right)$$

$k=2$  OctTree  
 $k=3$   
Quad Tree

