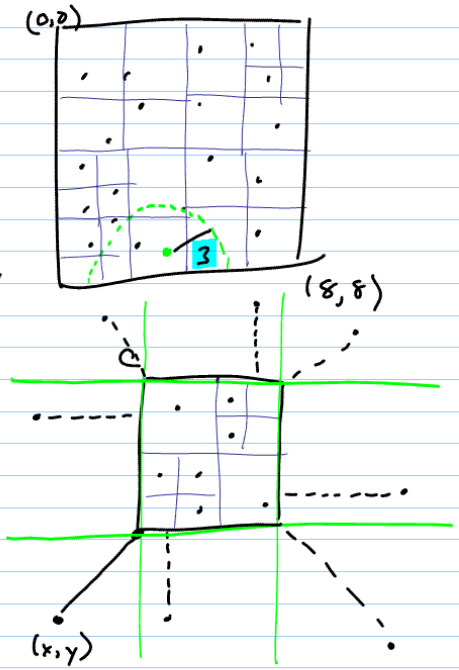
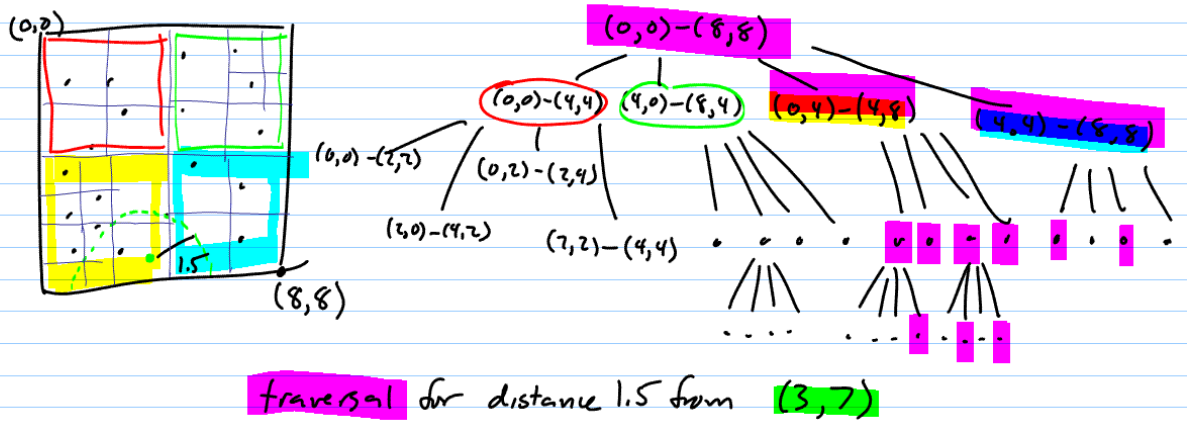
findClosePoints (x, y, d, n)

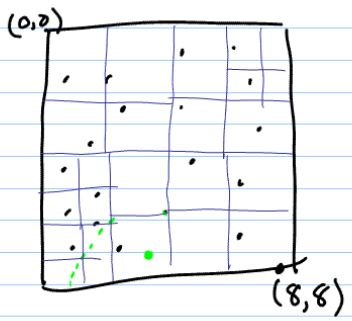
```
if n is empty
    return
if n has one point
    compute distance to point
    if < d then process
else
    for each child c
        compute distance from c to x,y
        if < d then
            findClosePoints(x,y,d,c)
```





traversal for distance 1.5 from (3,7)

nearestNeighbors (x, y, k)



use
uses priority queue

enqueue(item, pri)
removeMin()
peekMin()
isEmpty()

```
init priority queue Q        (priorities are dist to (x,y)
  Q.enqueue(root, 0)
init list l                  (k nearest points)

while !Q.isEmpty() and l.size() < k
    p ← Q.removeMin()
    if p is a point
        l.add(p)
    else
        for each child c of p
            Q.enqueue(c, distance from (x,y) to c)
```

Find 3 closest points to (3,7)

priority queue