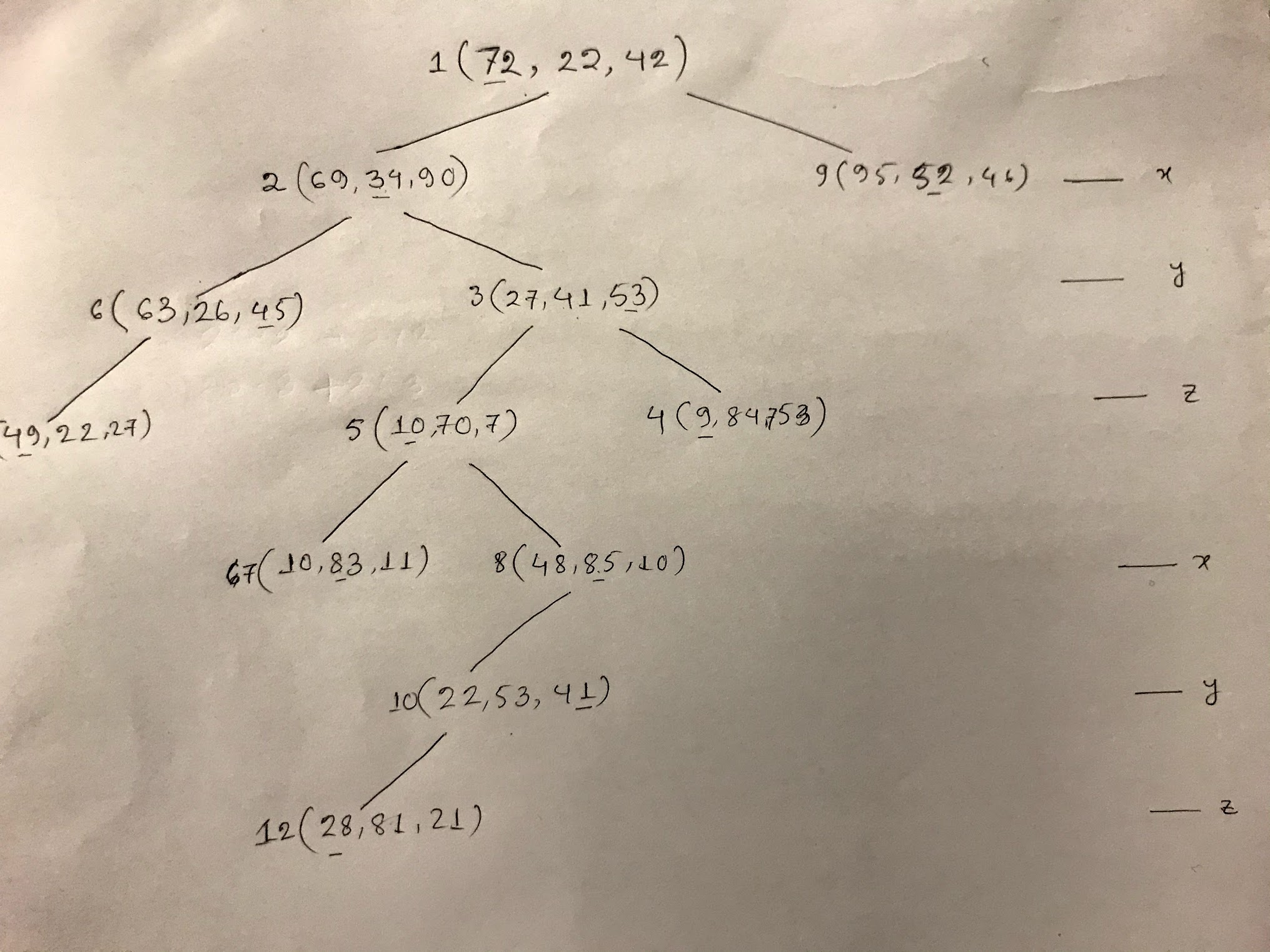
Problem 4:

a)



b)

Pseudocode: [1]

**KDTree** (leftNode, rightNode)

**Split** (value, dimention): implements **KDTree**

**Point** (v1, v2, ...): implements **KDTree**

*insert\_point*(**KDTree** T, List<of>**Point**):

// find the one with the largest variance

var max\_variance\_dimension

**for** all dimensions d:

sort points by dimension d

get two median points P1 **and** P2

get variance of P1 **and** P2:

avg = (P1.d.value + P2.d.value) / 2

variance = P1.d.value ~ avg

**if** variance is maximum:

update max\_variance\_dimension with current dimension

// find split point

sort by max\_variance\_dimension

get two median points P1 **and** P2

split\_value = (P1.max\_variance\_dimention.value + P1.max\_variance\_dimention.value) / 2

**if** T is null:

T = **new** **Split**(split\_value, max\_variance\_dimension)

**else** **if** split\_value <= T.leftNode.value:

T.leftNode = *insert\_point*(**T**.leftNode, List<of>**Point** that are lower than split\_value in max\_variance\_dimension)

**else** **if** split\_value > T.rightNode.value:

T.rightNode = *insert\_point*(**T**.rightNode, List<of>**Point** that are larger than split\_value in max\_variance\_dimension)

**return** T

Explanation:

1. At first as instructed determine the split dimension.
   1. Which is the one with the largest variance, we considered the difference of a median node to the average of two median nodes. [2]
   2. Run this for all dimension and find out the maximum variance.
2. The split value is calculated as the average of two median nodes of selected dimension in previous stage.
3. Build kd tree (left and right) using recursion. [3]

Reference:

[1] <https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/kdtrees.pdf>

[2] <https://en.wikipedia.org/wiki/Variance>

[3] Geometric algorithms and data structures: Prof. Suri (TU Darmstadt) <https://slideplayer.com/slide/4262586/>