Answer Key HW 6

1.1: False

1.2: True

2:

Proof: Since the tree is balanced, its height is $O(\log n)$. Since the constant factor will not really matter, it will simplify matters to assume that the height is exactly $\lg n$. Let us consider the case of a vertical line $x = x_0$. The horizontal case is symmetrical.

Consider a processed node which has a cutting dimension along x. The vertical line $x = x_0$ either stabs the left child or the right child but not both. If it fails to stab one of the children, then it cannot stab any of the cells belonging to the descendents of this child either. If the cutting dimension is along the y-axis (or generally any other axis in higher dimensions), then the line $x = x_0$ stabs both children's cells.

Since we alternate splitting on left and right, this means that after descending two levels in the tree, we may stab at most two of the possible four grandchildren of each node. (This is illustrated in Fig. 6.) In general each time we descend two more levels we double the number of nodes being stabbed. Thus, we stab the root node, at most 2 nodes at level 2 of the tree, at most 4 nodes at level 4, 8 nodes at level 6, and generally at most 2^i nodes at level $2i$.

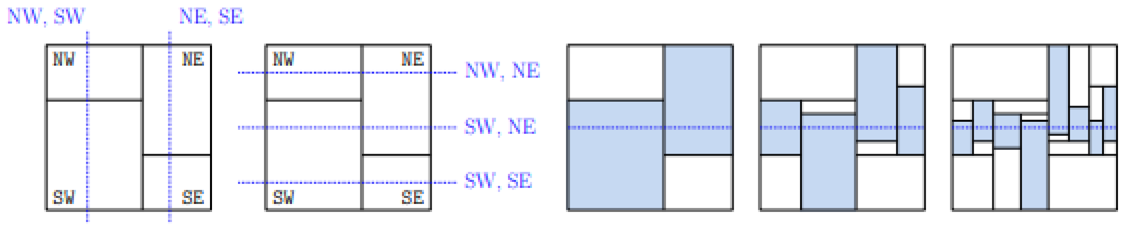


Fig. 6: An axis-parallel line in 2D can stab at most two out of four cells in two levels of the kd-tree decomposition. In general, it stabs 2^i cells at level $2i$.

Because we have an exponentially increasing number, the total sum is dominated by its last term. Thus, it suffices to count the number of nodes stabbed at the lowest level of the tree. If we assume that the kd-tree is balanced, then the tree has height of $h \approx \lg n$ (up to constant factors). The number of leaf nodes processed at the bottommost level is

$$2^{h/2} \approx 2^{(\lg n)/2} = (2^{\lg n})^{1/2} = n^{1/2} = \sqrt{n}.$$

This completes the proof.

We have shown that any vertical or horizontal line can stab only $O(\sqrt{n})$ cells of the tree. Thus, if we were to extend the four sides of Q into lines, the total number of cells stabbed by all these lines is at most $O(4\sqrt{n}) = O(\sqrt{n})$. Thus the total number of cells stabbed by the query range is $O(\sqrt{n})$, and hence the total query time is $O(\sqrt{n})$. Again, this assumes that the kd-tree is balanced (having $O(\log n)$ depth). If the points were inserted in random order, this will be true on average.

3: The worst case scenario is one where almost all points are equidistant from the query point.

4:

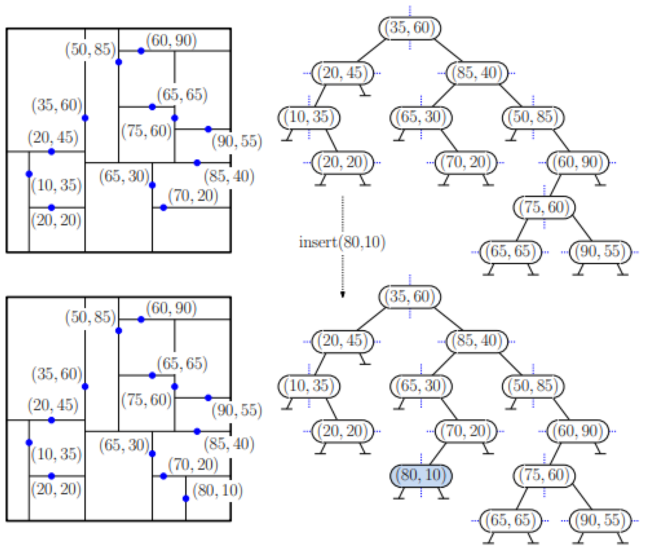
4.1) 10, 50, 28,70

4.2) 30, 0, 2, 3

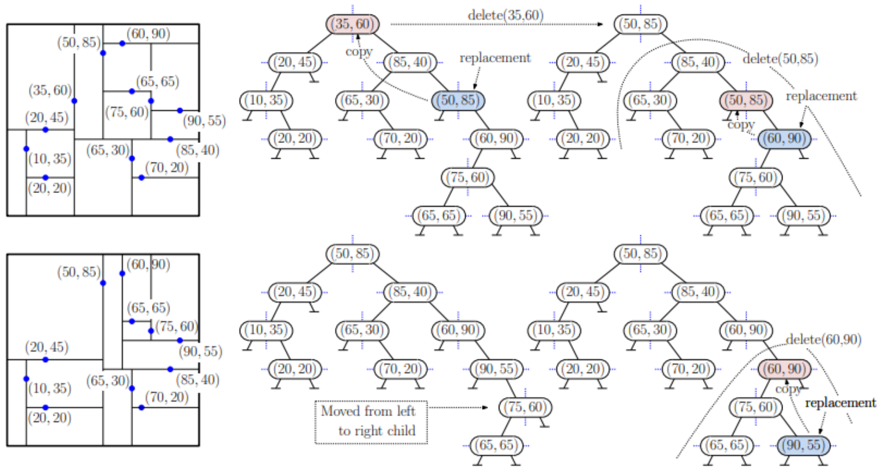
4.3) 10,60,30,50

4.4)16,51,30,0

4.5) 15,51,28,50



5.1:



5.2: