

522031910747+李若彬+hw6

简答（言之有理即可）

- 1.
- 2.

实践

- 1.
- 2.

简答（言之有理即可）

1. 在构造KD-Tree时，如何消除多点共垂直、共水平的退化情况？
2. KD-Tree相对于（二维）四分树、（三维）八分树，在什么情况下有什么优势？

1.

在构造 KD 树时，多点共垂直、共水平的退化情况是指在数据集中存在多个点具有相同的垂直或水平坐标。这种情况可能导致 KD 树的构建变得不平衡，进而影响查找的效率。

在 KD 树中，每个节点都对应着一个超矩形区域，该区域由一个切分超平面分割而成。如果数据集中存在多个点在同一切分维度上具有相同的坐标，则切分超平面可能会选择其中某个点作为切分点，这样会导致 KD 树的某一子树非常扁平，而另一子树非常丰满，从而影响 KD 树的平衡性。

为了避免这种情况，可以在构建 KD 树时采取一些策略：

1. **随机选择切分维度：** 在每一层递归构建 KD 树时，可以随机选择切分的维度，而不是按照固定的顺序选择。这样有助于避免在同一维度上出现多个点具有相同坐标的情况。
2. **处理重复点：** 如果在同一切分维度上存在多个点具有相同的坐标，可以选择一种策略来处理这些重复点，例如将它们视为一个点处理，或者在构建过程中避免选择这些重复点作为切分点。
3. **平衡性检查与重建：** 在构建完 KD 树后，可以进行平衡性检查，如果发现某些子树非常不平衡，可以考虑对其进行重建或调整，以提高 KD 树的平衡性。

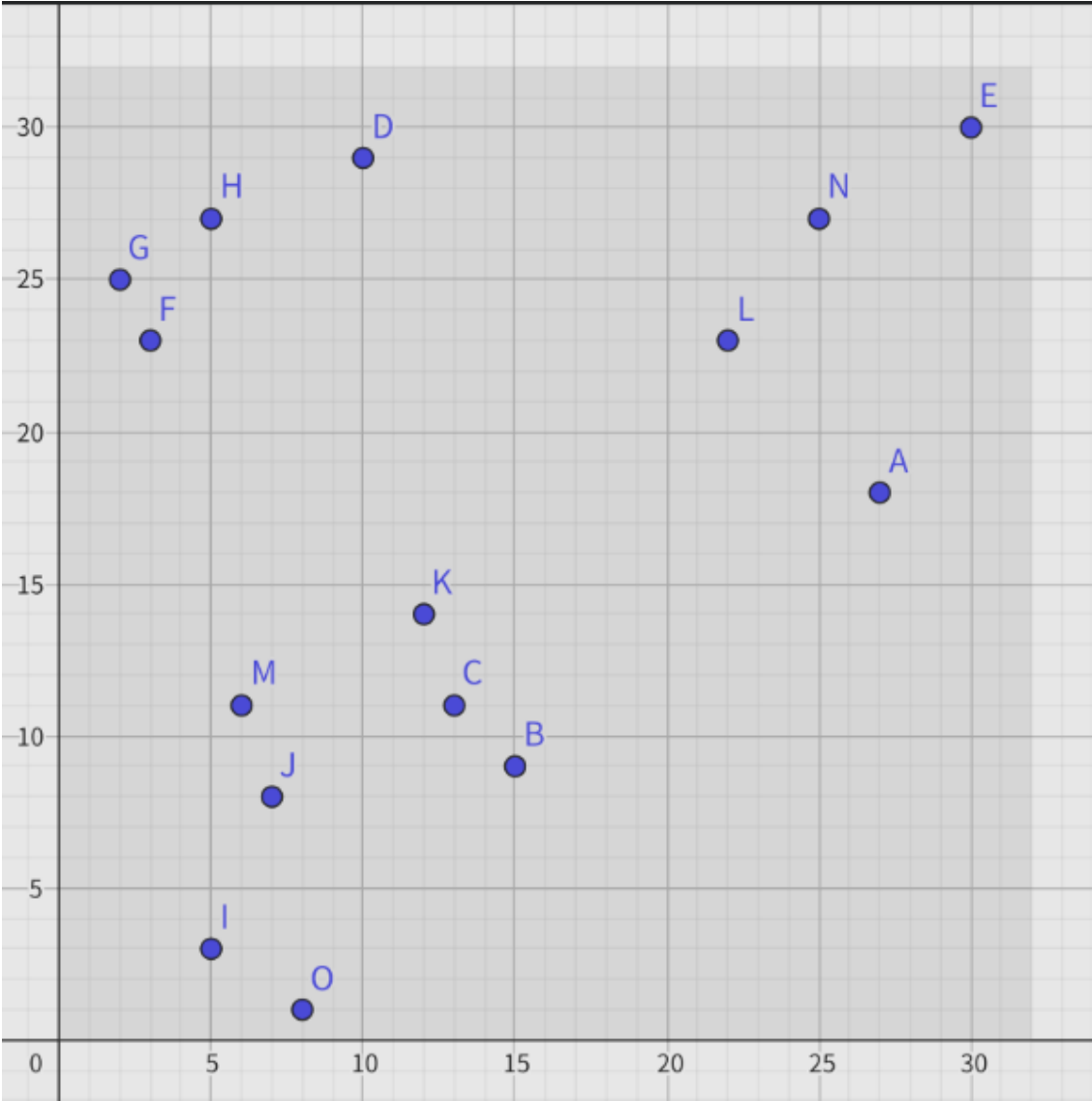
2.

KD 树相对于二维四分树和三维八分树在以下情况下具有优势：

1. **高维数据集：** KD 树在高维空间中的效率通常比四分树和八分树更高。在高维空间中，四分树和八分树的分支数量会迅速增加，导致树的结构变得非常复杂，而 KD 树在高维空间中仍然能够保持较好的性能。
2. **数据分布不均匀：** 当数据集的分布不均匀时，KD 树通常能够更好地适应这种情况。KD 树的构建过程中，根据数据点的实际分布选择切分维度，这使得 KD 树能够更好地适应不同维度上数据点的分布情况。
3. **查询效率：** 在某些情况下，KD 树的查询效率可能会比四分树和八分树更高。这取决于数据集的维度、分布情况以及具体的查询模式。

实践

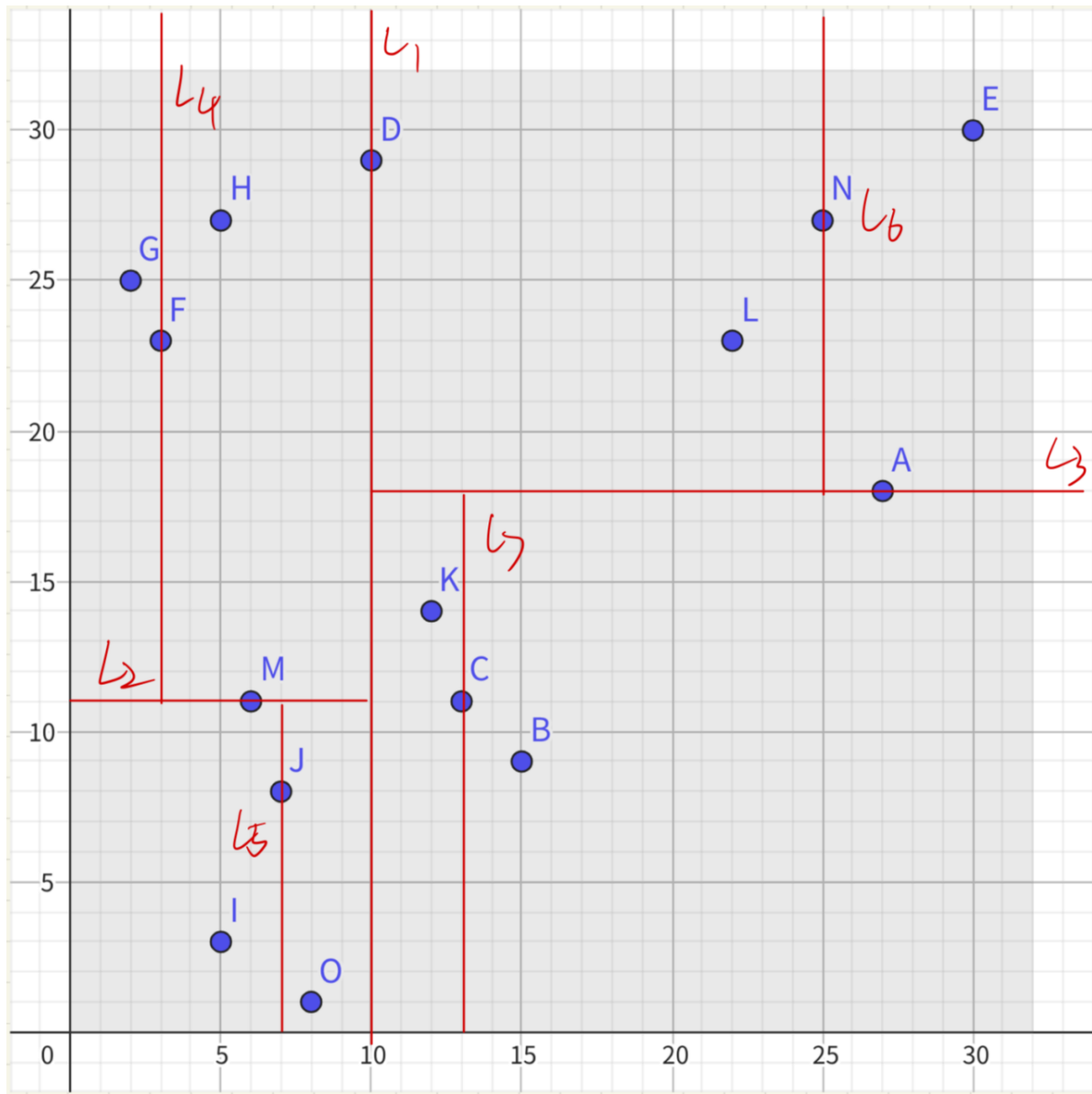
在以下 $[0,32] \times [0,32]$ 的二维空间中（灰色区域），有15个点。



请回答以下问题：

- 将这15个点建立KD-Tree（要求首先按照横轴划分）。
- 在构造的KD-Tree中，要查找离点 (20,20) 距离最近的点，请给出查找过程。

1.



2.

1. 首先根据D(10,29)判断选取在右侧分支
2. 再根据A(27,18)选取上侧分支
3. 其次再根据N(25,27)选取左侧分支
4. 最后选取分支中唯一的节点为L(22,23)