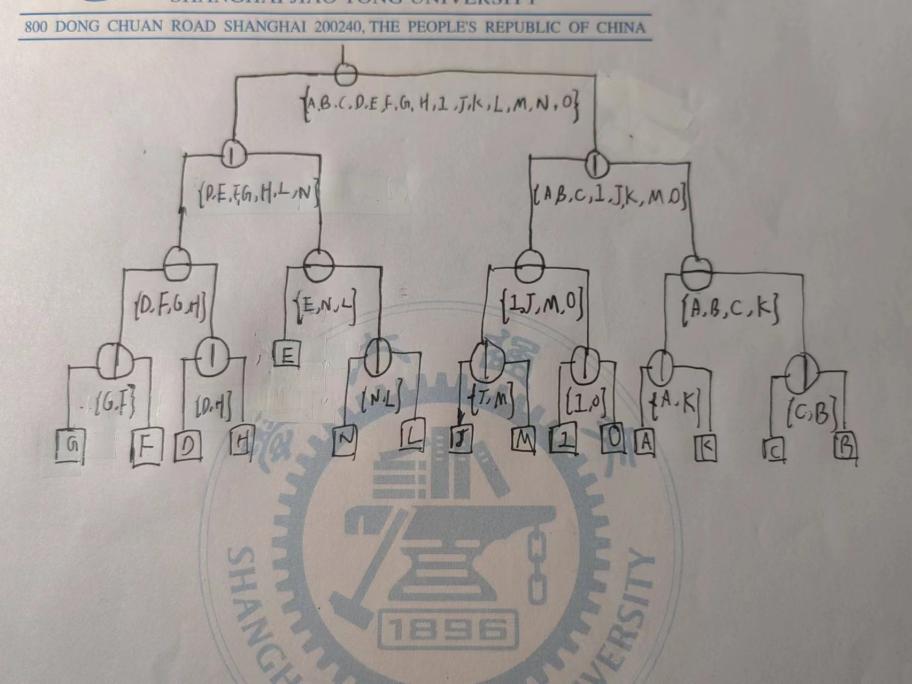
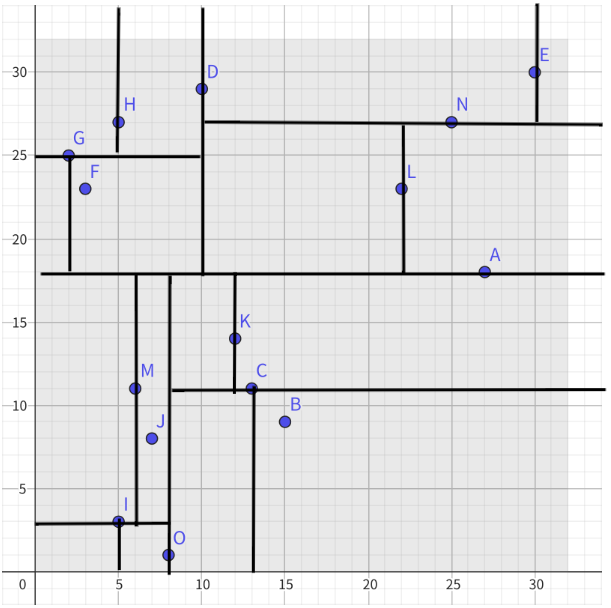
简答：

1、可以给每个数据点加一个权重，如果有多个数据在同一个直线上，优先选择这些数据中的权重的中位数所对应的数据点，然后权重比这个数据点高的认为在直线右边（下边），权重比这个数据点低的认为在直线左边（上边）

2、如果数据分布较均匀且树保持平衡（情况），则KD-Tree搜索效率相比四分树和八分树可能效率更高（优势）

实践：

1、



2、先从根节点进行搜索，找到所在的叶节点区域<L>和搜索路径<A,D,N,L>,此时最近节点设为L，然后回溯到N，但是N到（20,20）的距离比L到（20,20）大，此时最近节点依然是L，以（20,20）为圆心，和L之前的距离为半径，画圆，此时的圆与N所在的y=27无交点，继续向上回溯至D，但是D到（20,20）的距离比L到（20,20）大，此时最近节点依然是L，以（20,20）为圆心，和L之前的距离为半径，画圆，此时的圆与D所在的x=10无交点，继续回溯至节点A，但是A到（20,20）的距离比L到（20,20）大，此时最近节点依然是L，以（20,20）为圆心，和L之前的距离为半径，画圆，此时的圆与A所在的y=18有交点，此时移动到另一个子节点O，但是O到（20,20）的距离比L到（20,20）大，此时最近节点依然是L，以（20,20）为圆心，和L之前的距离为半径，画圆，此时的圆与O所在的x=8无交点，回溯结束，结果为L。