**EXP1:**

(-5.429850155, 0.807567048, -0.398216823) query point has number of 5 nearest neighbors of an eps value of 0.05. whatever search method using (Linear or KD-Tree) the nearest neighbors will always have same list of neighbors.

(-12.97637373, 5.09061138, 0.762238889) query point has number of 2 nearest neighbors of an eps value of 0.05.

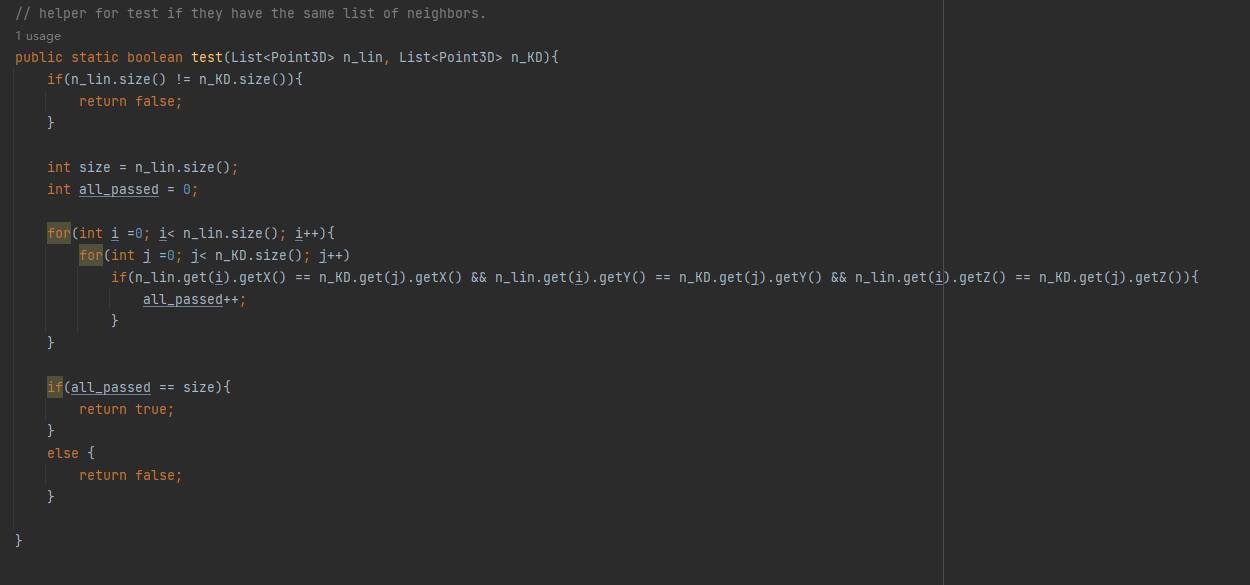
(-36.10818686, 14.2416184, 4.293473762) query point has number of 1 nearest neighbors of an eps value of 0.05.

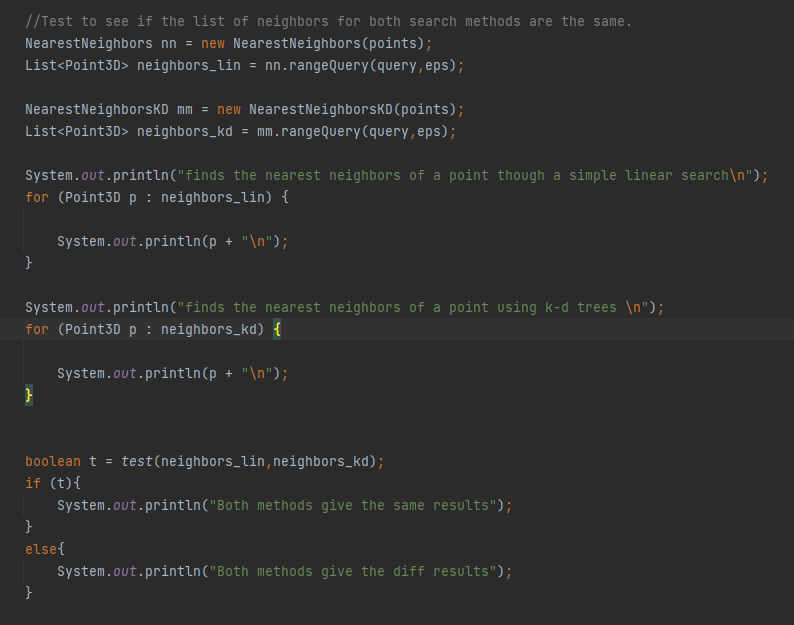
(3.107437007, 0.032869335, 0.428397562) query point has number of 17 nearest neighbors of an eps value of 0.05.

(11.58047393, 2.990601868, 1.865463342) query point has number of 2 nearest neighbors of an eps value of 0.05.

(14.15982089, 4.680702457, -0.133791584) query point has number of 2 nearest neighbors of an eps value of 0.05.

**Test**:

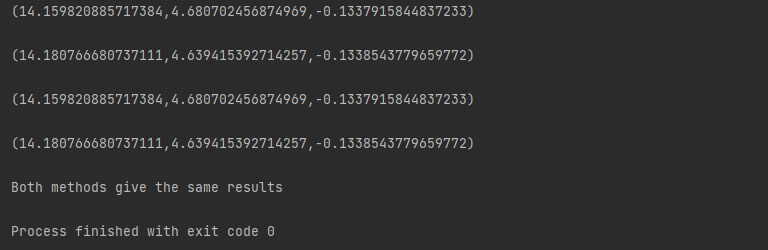




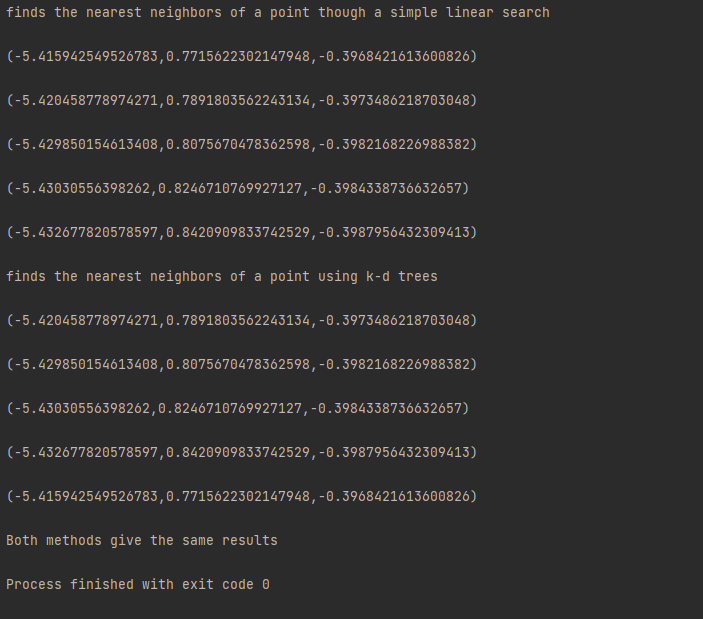
Above methods used to test

Some tests example：

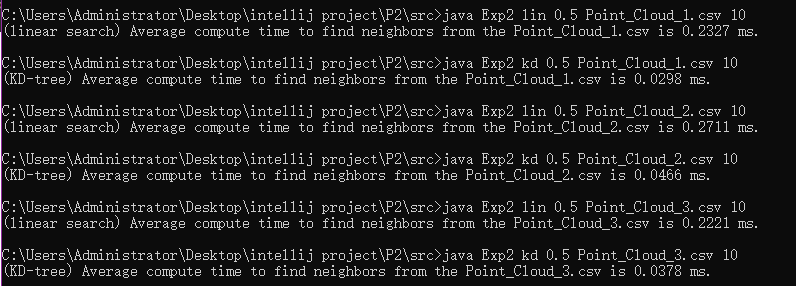
(14.15982089, 4.680702457, -0.133791584) query point



(-5.429850155, 0.807567048, -0.398216823) query point



**Exp2:**



For all the files, the run time for the linear search is always the slower one, and KD-tree is always way faster, this is what is expected. Note that file 3 has more points than file 1 but the linear search for file 3 is a bit faster than file 1, because It depends on your computer's CPU, more than that execution time is different each time for all, but not much difference.

**Exp3:**

**Linear search:**

****





**KD-TREE:**

****





From the results, it is obvious that KD-tree is faster and takes less time. This is the expected result and what is wanted. Judging from the results, the gap is still very large, and KD-tree is almost twice as fast. In terms of time plus reading files and finding clusters, etc., these times seem reasonable. Finally, using KD-tree can reduce more costs.