Random Geographic Graph

Brute Force & K-D Tree Algorithm

N = 1000, Avg Deg = 10

```
Brute Force:
Consuming time: 0.125 seconds
K-D Tree:
Consuming time: 0.044999957 seconds
```

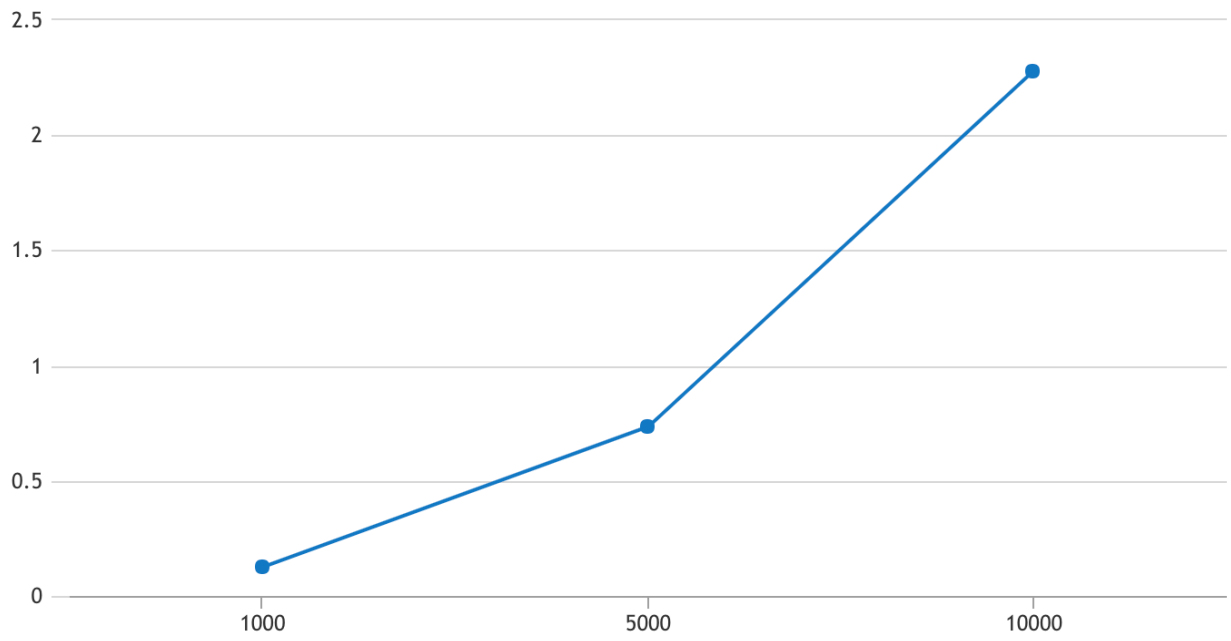
N=5000, Avg Deg = 10

```
Brute Force:
Consuming time: 0.73599994 seconds
K-D Tree:
Consuming time: 0.21500003 seconds
```

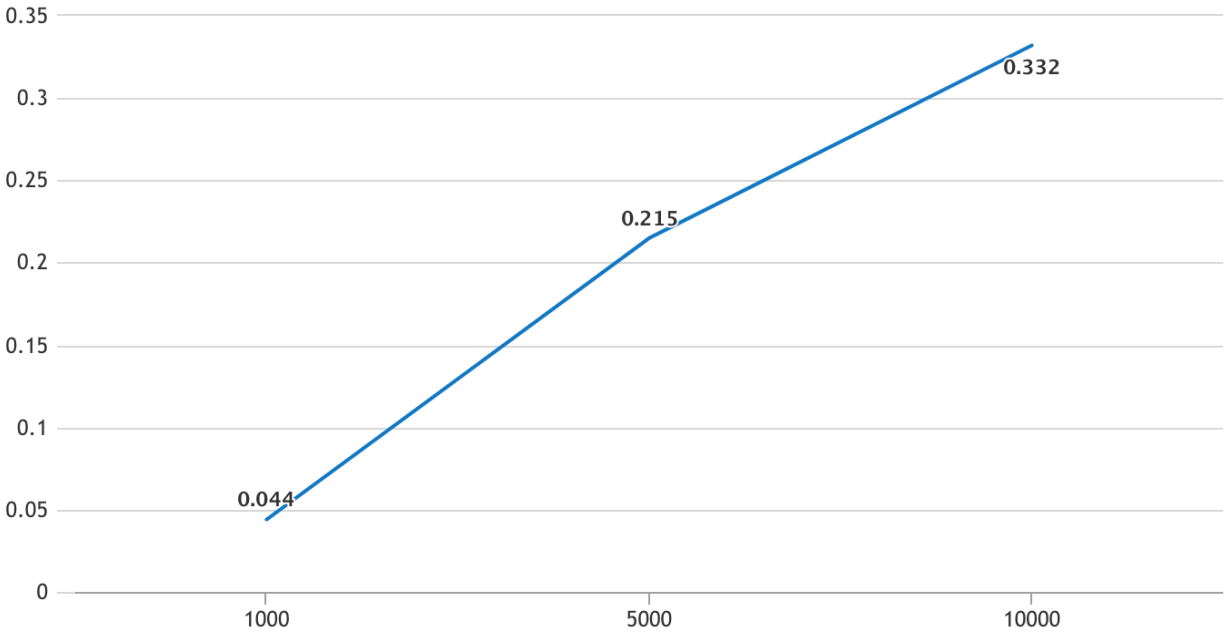
N=10000, Avg Deg = 10

```
Brute Force:
Consuming time: 2.281 seconds
K-D Tree:
Consuming time: 0.33299994 seconds
```

Brute Force:



K-D Tree:



Something I found is interesting before I use K-D Tree:

After went through and researched varies of popular implementations of k-d tree, the most different idea is how to maintain the balancing of k-d tree while in the building stage, which can improve efficiency of all operations later on. However, in this case which is RGG, all vertices are already distributed uniformly, which means we don't need to consider the balancing issue. Besides, we don't need to consider the sequence and the dimension, which is from where and how to split the space either.