# Clustering

- Programming assignment #3

2013011060
양승호

## 1. Environment

Ubuntu Linux 18.04
Python 3.6.7

## 2. How to run

$ python3 clustering.py [input_file_name] [number_of_clustering] [Eps] [minPts]
(example: python3 clustering.py input2.txt 5 2 7)

## 3. Algorithm Summary

### - DBSCAN

1. Arbitrary select a starting point.(I used the first object of the input file)
2. Get the density-reachable points from the prior selected point
3. Merge the new cluster with prior cluster if the core points are density-reachable each other.
4. Repeat until all the points are checked at least once.

# 4. Details

```
23      for ob_1 in objects:
24          flag_combine = False
25          tmp = [ob_1[0]]
26          for ob_2 in objects:
27              if ob_2[0] == ob_1[0]:
28                  continue
29              if Distance((float(ob_1[1]), float(ob_1[2])), (float(ob_2
    [1]), float(ob_2[2]))) <= eps:
30                  tmp.append(ob_2[0])
31
32          if len(tmp) < minpts:
33              continue
34
35          for idx in range(len(results)):
36              if ob_1[0] in results[idx]:
37                  flag_combine = True
38                  break
39
40          if flag_combine:
41              results[idx] += tmp
42              results[idx] = list(set(results[idx]))
43              core_points[idx].append(ob_1[0])
44          else:
45              results.append(tmp)
46              core_points.append([ob_1[0]])
```

- main function of DBSCAN
- Iterating in each objects of inputfile, find all points which are closer than the range of epsilion, and check the condition whether the number of satisfying points is more than minPts.
- Check whether the core point is already in the other cluster.
- If so, then combine the clusters and repeat the process.

```
3 def Distance(pt1, pt2):
4     return ((pt1[0] - pt2[0])**2 + (pt1[1] - pt2[1])** 2)**0.5
```

- return the distance between two points