

K개의 클러스터, D개의 차원, N개의 데이터를 이용해 K-means 클러스터링을 진행했습니다.

동일 입력값을 2회씩 반복해서 입력해보겠습니다.

(각 클러스터내의 데이터 수를 쉽게 파악하기위한 코드를 따로 넣었습니다.)

풀이방법:

구조체를 만들어서 x, y... n차원 값을 담도록 하고 이를 배열로 만들어 인덱스를 통해 호출하도록 했습니다.

input파일에서 k d n 을 가져오고 한줄씩 입력을 받아 각각의 구조체에 할당합니다.

입력된 값에서 무작위로 k개를 뽑아 클러스터값에 대입합니다.

이후, 유클리드 거리를 이용해 클러스터링을 진행하고, 이를 기반으로 새롭게 클러스터를 지정. 이를 반복하고 변경되지 않을때 종료되도록 코딩하였습니다.

첫번째 입력 파일:

1-1 아웃풋 파일:

1-2 아웃풋 파일:

```
5 3 30
388 198 210
476 667 294
218 357 778
417 323 169
650 600 97
637 494 853
92 73 575
328 481 387
828 112 457
151 139 991
346 251 406
936 49 52
309 700 273
859 31 595
966 121 278
320 330 349
713 225 158
47 257 2
701 434 863
37 280 611
576 733 633
983 141 628
935 754 750
484 146 249
354 912 124
334 383 435
814 219 993
375 174 933
46 21 386
604 334 480
```

```
iter: 1
1 3 7 8 10 11 16 20 24 26 28 29 30 total: 13
6 9 14 15 19 22 27 total: 7
2 4 5 12 13 17 25 total: 7
18 total: 1
21 23 total: 2
iter: 2
1 3 7 8 10 11 16 20 24 26 28 29 total: 12
9 14 15 22 27 30 total: 6
2 4 5 12 13 17 25 total: 7
18 total: 1
6 19 21 23 total: 4
iter: 3
3 7 10 11 16 20 26 28 29 total: 9
9 12 14 15 22 27 30 total: 7
1 2 4 5 8 13 17 24 25 total: 9
18 total: 1
6 19 21 23 total: 4
iter: 4
3 7 10 11 20 28 29 total: 7
9 12 14 15 22 30 total: 6
1 2 4 5 8 13 16 17 24 25 26 total: 11
18 total: 1
6 19 21 23 27 total: 5
```

```
iter: 1
3 7 10 20 28 29 total: 6
1 4 8 9 11 12 14 15 16 17 18 24 26 30 total: 14
6 19 21 22 27 total: 5
23 total: 1
2 5 13 25 total: 4
iter: 2
3 7 10 20 28 29 total: 6
1 4 9 11 12 14 15 16 17 18 24 26 30 total: 13
6 19 21 22 27 total: 5
23 total: 1
2 5 8 13 25 total: 5
iter: 3
3 7 10 20 28 29 total: 6
1 4 9 11 12 15 16 17 18 24 26 30 total: 12
6 14 19 22 27 total: 5
21 23 total: 2
2 5 8 13 25 total: 5
iter: 4
3 7 10 20 28 29 total: 6
1 4 11 12 15 16 17 18 24 26 30 total: 11
6 9 14 19 22 27 total: 6
21 23 total: 2
2 5 8 13 25 total: 5
iter: 5
3 7 10 20 28 29 total: 6
1 4 11 12 15 16 17 18 24 26 30 total: 11
9 14 22 27 total: 4
6 19 21 23 total: 4
2 5 8 13 25 total: 5
iter: 6
3 7 10 20 28 29 total: 6
1 4 11 12 16 17 18 24 26 30 total: 10
9 14 15 22 27 total: 5
6 19 21 23 total: 4
2 5 8 13 25 total: 5
iter: 7
3 7 10 20 28 29 total: 6
1 4 11 16 17 18 24 26 30 total: 9
9 12 14 15 22 27 total: 6
6 19 21 23 total: 4
2 5 8 13 25 total: 5
iter: 8
3 7 10 20 28 29 total: 6
1 4 8 11 16 17 18 24 26 30 total: 10
9 12 14 15 22 total: 5
6 19 21 23 27 total: 5
2 5 13 25 total: 4
iter: 9
3 7 10 20 28 29 total: 6
1 4 8 11 16 18 24 26 30 total: 9
9 12 14 15 17 22 total: 6
6 19 21 23 27 total: 5
2 5 13 25 total: 4
```

두번째 입력 파일:

2-1 아웃풋 파일:

2-2 아웃풋 파일:

4 2 30  
79 56  
6 52  
4 46  
54 79  
45 29  
18 45  
53 68  
62 67  
99 74  
17 94  
92 60  
21 84  
76 78  
59 70  
56 37  
81 69  
71 7  
83 66  
22 87  
54 92  
20 11  
47 71  
99 43  
31 26  
29 56  
28 90  
66 47  
42 3  
16 86  
94 5

```
iter: 1
2 3 6 10 12 19 25 26 29 total: 9
1 9 11 16 17 18 23 30 total: 8
5 7 8 15 21 22 24 27 28 total: 9
4 13 14 20 total: 4
iter: 2
2 3 6 10 12 19 25 26 29 total: 9
1 9 11 16 18 23 30 total: 7
5 15 17 21 24 27 28 total: 7
4 7 8 13 14 20 22 total: 7
iter: 3
2 3 6 10 12 19 25 26 29 total: 9
1 9 11 16 18 23 27 30 total: 8
5 15 17 21 24 28 total: 6
4 7 8 13 14 20 22 total: 7
```

```
iter: 1
1 4 8 9 11 13 14 16 18 23 27 30 total: 12
10 12 19 20 26 29 total: 6
2 3 6 21 24 28 total: 6
5 7 15 17 22 25 total: 6
iter: 2
1 8 9 11 13 16 18 23 27 30 total: 10
4 10 12 19 20 26 29 total: 7
2 3 6 21 24 28 total: 6
5 7 14 15 17 22 25 total: 7
iter: 3
1 9 11 13 16 17 18 23 30 total: 9
10 12 19 20 26 29 total: 6
2 3 5 6 21 24 28 total: 7
4 7 8 14 15 22 25 27 total: 8
iter: 4
1 9 11 16 17 18 23 30 total: 8
10 12 19 20 26 29 total: 6
2 3 5 6 21 24 25 28 total: 8
4 7 8 13 14 15 22 27 total: 8
```