

Assignment #2

|  |  |
| --- | --- |
| 과목명. | 인공지능 |
| 담 당. | 최 용 석 교수님 |
| 제출일. | 2020년 11월 20일 |
| 공과대학 | 컴퓨터소프트웨어학부 |
| 학 번 | 2016025969 |
| 이 름. | 정지훈 |

****

**목차.**

**1. 코드 설명**

**- 코드 실행 방법**

**- 코드 동작**

**2. 함수 설명**

**- Cosine similarity**

**- Disjoint Set**

**- 공통 부분**

**- Single**

**- Complete**

**- Average**

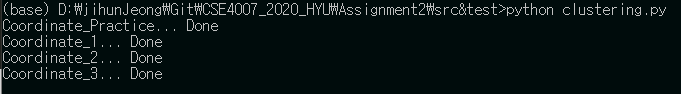
**3. 실험 결과**

**- Clustering**

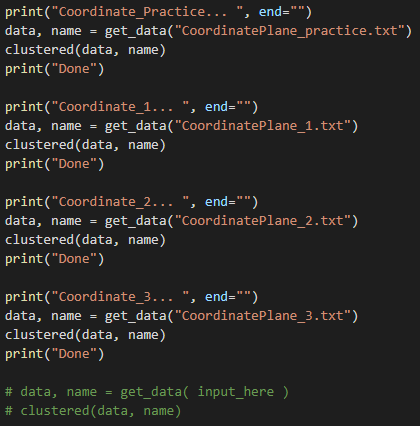
**- Span**

**1. 코드 설명**

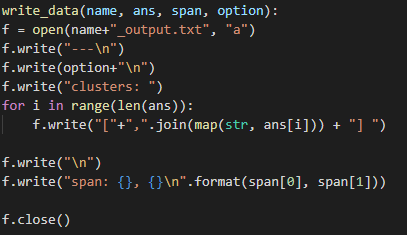
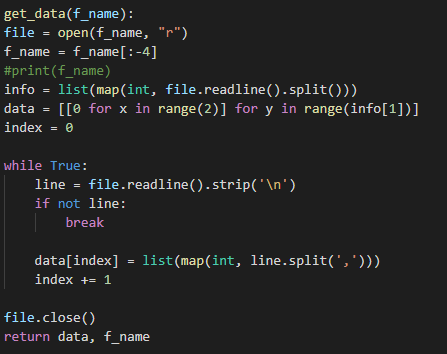
**- 코드 실행 방법**

****

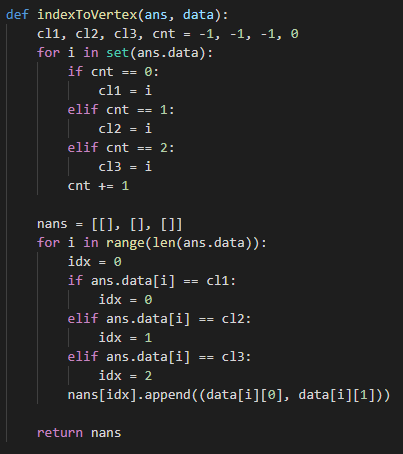
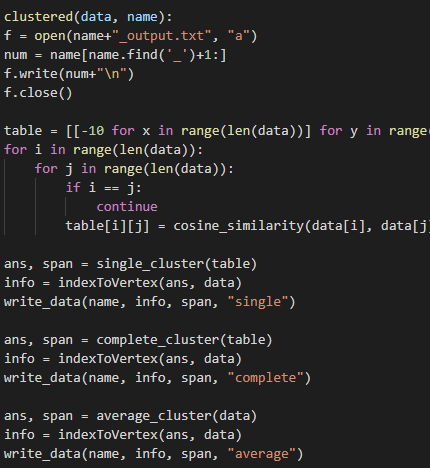
제출한 파일을 기존 compiler를 이용해서 실행시키면 과제의 예제 practice와 주어진 Coordinate 1~3까지의 clustering을 진행합니다.

다른 미로를 탐색하고 싶을 때 주석 처리한 input\_here 부분을 통해 원하는 좌표를 진행할 수 있습니다. 단 file 명과 확장자로 인해 py file과 같은 위치에 있어야 하며 txt 파일이어야 합니다.

**- 코드 동작**



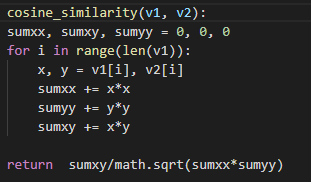
Get data function을 통해 주어진 txt 점의 좌표를 2차원 배열에 넣어줍니다. Write data function을 통해 Clustering 결과와 경계 Span을 ouput.txt에 넣어줍니다. Option을 통해 해당 결과가 어떤 type인지를 정해줍니다.

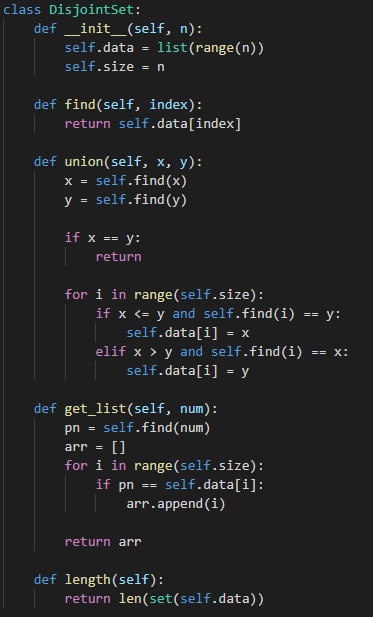


Get data를 통해 얻은 정보를 가지고 Clustering을 진행하는 clustered function입니다. Data를 가지고 오른쪽에 보이는 cosine similarity를 이용해 두 벡터의 유사도를 구한 값을 table에 넣어줍니다. 이 유사도를 가지고 3가지의 clustering을 진행합니다. Clustering에서는 점을 index로 표시해 구하기 때문에 나온 결과를 다시 점으로 바꿔주는 indexToVertex function을 사용합니다. 이 때 disjoint Set에 의해 3개의 집단이 표현이 됩니다.

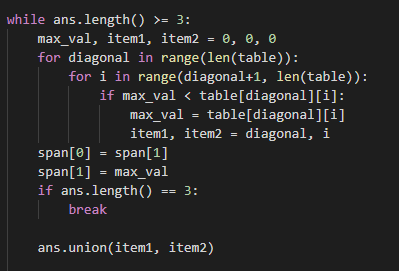
**2. 함수 설명**

**- Cosine similarity**

****코사인 유사도의 코드입니다. 두 벡터의 내적에 벡터의 크기를 나눈 코사인 값을 이용해 유사도를 측정합니다. 두 벡터가 같을 때는 1의 값을 정반대를 가질 때에는 -1의 값을 가지게 됩니다.

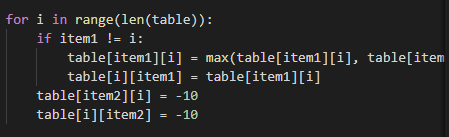
**- Disjoint Set**

Clustering 된 점들을 표현하기 위한 Disjoint Set 자료 구조입니다. 초기에는 각 점들의 index로 초기화를 하고 만약 두 점이 속한 집단이 합쳐져 하나의 clustering이 될 때 Union 함수를 통해 두 집단을 합쳐 줍니다. 한 개의 점이 들어왔을 때 그 점이 속한 집단의 점들을 가지고 오는 get\_list 함수도 있습니다. 집단이 총 몇 개 있는지 lengh method의 set 함수를 사용해 알 수 있습니다.

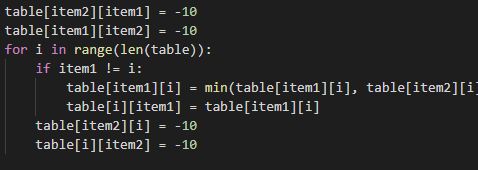
**- 공통 부분**

유사도가 담겨 있는 table에서 다음 집합으로 span 되기 위해 table에서 max 값과 해당 index를 얻습니다. 유사도의 경계를 알아야 하기에 경계를 담는 span 배열에 max\_val를 담습니다. 그렇게 얻은 두 index는 union function을 통해 하나의 clustering으로 합쳐지게 됩니다.

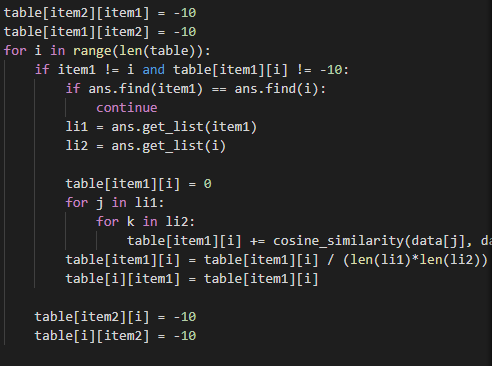
**- Single clustering**



Single clustering은 두 집단이 합쳐진 다음 기존 두 집단의 유사도 값 중에 높은 값이 합쳐진 하나의 집단의 대표 값이 되므로 max 함수를 써 그 값을 넣어줍니다. 두 집단이 하나의 집단이 되었기에 나머지 한 집단은 더 이상 값이 없다는 의미인 -10으로 모든 값을 바꿔줍니다.

**- Complete clustering**

Complete Clustering은 두 집단이 합쳐질 때 두 집단에서 다른 집단의 코사인 유사도 값 중에 작은 값이 새로운 집단의 유사도로 초기화가 되므로 min의 함수를 써서 합쳐지는 새로운 집단의 값으로 초기화 해줬습니다. 합쳐지고 난 뒤 나머지 집단은 더 이상 사용하지 않는 다는 의미로 -10으로 바꿔줍니다.

**- Average clustering**

Average Clustering은 두 집단이 합쳐질 때 합쳐진 집단의 모든 점에서 다른 집단의 모든 점 사이 유사도의 평균으로 두 집단의 유사도를 나타내기에 합쳐진 집단과 기존 다른 집단 안에 있는 모든 점을 get list로 얻어와 코사인 유사도의 평균을 새로운 값으로 넣어줍니다.

**3. 실험 결과**

**- Clustering**

(첨부 파일 참조)

**- Span**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Practice | Coordinate\_1 | Coordinate\_2 | Coordinate\_3 |
| Single Upper | 0.9910 | 0.7363 | 0.9258 | 0.9510 |
| Single Lower | 0.9894 | 0.7071 | 0.9194 | 0.8871 |
| Complete Upper | 0.9910 | 0.7363 | -0.7050 | -0.5952 |
| Complete Lower | 0.9193 | 0.6000 | -0.7441 | -0.9927 |
| Average Upper | 0.9910 | 0.7363 | 0.1542 | 0.1331 |
| Average  Lower | 0.9584 | 0.6536 | 0.0795 | -0.0728 |

*(소수 5번째 자리에서 반올림한 값)*