알고리즘 응용
- week12 행렬 바꾸기 -

제출일자	2021.05.30
분 반	00
이 름	강인한
학 번	201701969

1	코드 설명

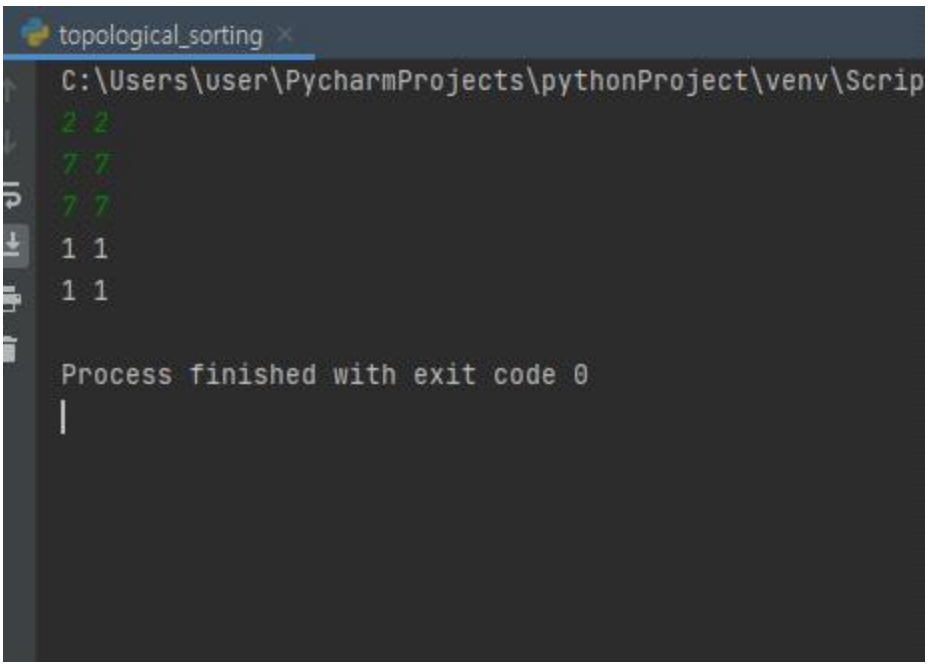
```
while renum<n*m:
    maxList=[]
    for i in range(n):
        array2= array[:,i]
        minValue= min(array2) #행 중에서 가장 작은 값 찾는 과정
        cnt= array2.tolist().count(minValue)
        j= array2.tolist().index(minValue)  # 행중에서 가장 작은 값의 개수
        if minValue == min(array[j,:]) and minValue != 1000000001: #행에서 가장 작은 값이 열에서도 가장 작다면 결과 list에 추가
            result[j,i]=num
            maxList.append((j,i))
            renum+=1
        if cnt>=2:
            while cnt>1:  # 동일한 값을 처리해 주기 위한 코드
                array2 = array[j+1:,i]
                minValue = min(array2)
                cnt = array2.tolist().count(minValue)
                j = array2.tolist().index(minValue)+j+1
                if minValue == min(array[j,:]) and minValue != 1000000001:
                    result[j,i] = num
                    maxList.append((j, i))
                    renum+=1
                cnt-1

    for k in range(len(maxList)):# 결과 list에 들어간 인덱스의 값을 1000000001로 채운다.
        array[maxList[k][0],maxList[k][1]]= maxNum
    num+=1
```

문제를 해결하기 위해 위상정렬의 방법을 이용하였다. 위상정렬과 다른 점으로는 위상정렬에서는 선행 노드가 없을 때(input이 0인 경우) 노드가 차례대로 실행되었는데 여기서는 값이 가장 작은 경우 노드가 실행되게 된다. 정확히 말하자면 행에서 가장 작은 값이 열중에서도 가장 작다면 조건을 만족하게 되고 1부터 차례로 입력된다. 처리된 값에는 1000000001이라는 값을 넣어 다음 탐색에서는 최솟값이 되지 않도록 한다.

행에서 min값을 찾고 min값의 개수를 구하여 모든 최솟값에 대하여 같은 숫자를 부여해주어야 한다. 아래의 while문을 사용한 이유이다.

2	실행 결과


topological_sorting
C:\Users\user\PycharmProjects\pythonProject\venv\Scripts\python.exe C:/Us
4 3
20 -21 14
-19 4 19
22 -47 24
-19 4 19
4 2 3
1 3 4
5 1 6
1 3 4
Process finished with exit code 0


topological_sorting
C:\Users\user\PycharmProjects\pythonProject\venv\Scrip
2 2
7 7
7 7
1 1
1 1
Process finished with exit code 0