

# 알고리즘 실습 보고서

- MCM & BF -

전공 : 컴퓨터공학과

분반 : 05반

학번 : 201502085

이름 : 이규봉

## 1. 실행 환경

과제의 코드를 테스트한 환경은 아래와 같습니다. 따로 사용한 라이브러리는 없습니다.

- Windows 10
- Pycharm 2019.02
- Python 3.7.4

## 2. 과제 설명

- DP를 활용한 벨만 포드 알고리즘 구현 및 MCM 구현

## 3. 문제 해결 방법

- 문제를 해결하기 위해 자신이 사용한 방법, 아이디어에 대한 설명  
(구현한 코드에 대한 설명 등)

벨만 포드 알고리즘의 경우 DP를 활용해 for문을 돌며 이 중 for문에서 각 점들끼리의 거리를 계산하고 업데이트 합니다. 그리고 배열 dp에 저장된 거리 값을 업데이트 하고 outer for loop에서 dp의 각 값들을 출력합니다. 시간 복잡도는  $O(n^3)$ 입니다.

MCM의 경우, 배열 M에 행렬 곱에 필요한 최소의 곱셈 횟수를 저장해 놓고, DP를 활용해 행렬 곱셈의 횟수가 제일 적어지는 M 값을 선택하여 업데이트 합니다. 최소 곱셈의 수는 mcm 메서드에서 계산하는 것이 아니라, mcm으로 리턴된 배열을 받아, 반복문으로 최소값을 구하여 출력하도록 했습니다. 마찬가지로 시간 복잡도는  $O(n^3)$ 입니다.

## 4. 결과 화면

bellmanFord 알고리즘 data 1

```
C:\Algorithm-HW\venv\Scripts\python.exe C:/Algorithm-HW/Assign09/bellmanFord.py
-----iteration 0-----
Update distance of 1 from inf to 6
Update distance of 2 from inf to 7
Update distance of 3 from inf to 11
Update distance of 4 from inf to 2
Update distance of 3 from 11 to 4
Update distance of 1 from 6 to 2
iteration 0 distance : [0, 2, 7, 4, 2]
-----iteration 1-----
Update distance of 4 from 2 to -2
iteration 1 distance : [0, 2, 7, 4, -2]
-----iteration 2-----
iteration 2 distance : [0, 2, 7, 4, -2]
-----iteration 3-----
iteration 3 distance : [0, 2, 7, 4, -2]
-----iteration 4-----
iteration 4 distance : [0, 2, 7, 4, -2]

Process finished with exit code 0
```

bellmanFord 알고리즘 data 2

```
C:\Algorithm-HW\venv\Scripts\python.exe C:/Algorithm-HW/Assign09/bellmanFord.py
-----iteration 0-----
Update distance of 1 from inf to 4
Update distance of 2 from inf to 3
Update distance of 2 from 3 to 0
Update distance of 0 from 0 to -2
iteration 0 distance : [-2, 4, 0]
-----iteration 1-----
Update distance of 1 from 4 to 2
Update distance of 2 from 0 to -2
Update distance of 0 from -2 to -4
iteration 1 distance : [-4, 2, -2]
-----iteration 2-----
Update distance of 1 from 2 to 0
Update distance of 2 from -2 to -4
Update distance of 0 from -4 to -6
iteration 2 distance : [-6, 0, -4]
The graph has negative cycle

Process finished with exit code 0
```

MCM

```
C:\Algorithm-HW\venv\Scripts\python.exe C:/Algorithm-HW/Assign09/matrixChainMultiplication.py
0 15750 7875 9375 11875 15125
None 0 2625 4375 7125 10500
None None 0 750 2500 5375
None None None 0 1000 3500
None None None None 0 5000
None None None None None 0

최소 곱셈의 수 : 750

Process finished with exit code 0
```

## 5. 느낀점 및 고찰

벨만포드 알고리즘 구현이 생각보다 힘들었지만, 전체적으로 적당한 난이도였던 것 같습니다.