

[실습 및 과제 4]

Youn-Hee Han

LINK@KOREATECH

<http://link.koreatech.ac.kr>

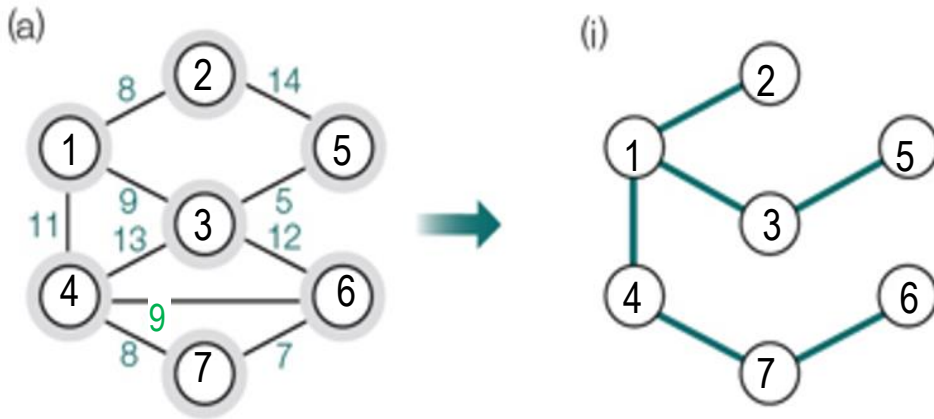
제시 문제

◆ [문제 1] 최소 비용 신장 트리 알고리즘 구현 (1/2)

- 제시된 [minimum_spanning_tree.py](#)의 빈 라인을 채우세요.
 - Graph 클래스 정의 소스 완벽 분석 및 이해 필요
 - DisjointSet 클래스 정의 소스 완벽 분석 및 이해 필요
 - union 함수 내 빈 라인 구현 필요
 - SpanningTree 클래스 정의 소스 완벽 분석 및 이해 필요
 - extract_min 함수 내 빈 라인 구현 필요
 - prim 함수 내 빈 라인 구현 필요
 - kruskal 함수 내 빈 라인 구현 필요
- 주어진 코드 수정 불가
 - 주어진 변수 이름 등도 수정 없이 그대로 사용
- 다음 페이지의 [숙제 코드 수행결과]와 동일한 수행 결과 출력 필요

제시 문제

◆ [문제 1] 최소 비용 신장 트리 알고리즘 구현 (2/2)



[주어지는 그래프]

```
adjacency_list = {  
    '1': {'2': 8, '3': 9, '4': 11},  
    '2': {'1': 8, '5': 14},  
    '3': {'1': 9, '4': 13, '5': 5, '6': 12},  
    '4': {'1': 11, '3': 13, '6': 9, '7': 8},  
    '5': {'2': 14, '3': 5},  
    '6': {'3': 12, '4': 9, '7': 7},  
    '7': {'4': 8, '6': 7}  
}
```

[숙제 코드 수행결과]

[Graph] number of nodes: 7, number of edges: 10

{('3', '5', 5), ('3', '6', 12), ('6', '7', 7), ('2', '5', 14), ('1', '4', 11), ('1', '3', 9), ('3', '4', 13), ('4', '6', 9), ('4', '7', 8), ('1', '2', 8)}

*** Prim Solution ***

1 - 2

1 - 3

1 - 4

3 - 5

7 - 6

4 - 7

*** Kruskal Solution ***

3 - 5

6 - 7

4 - 7

1 - 2

1 - 3

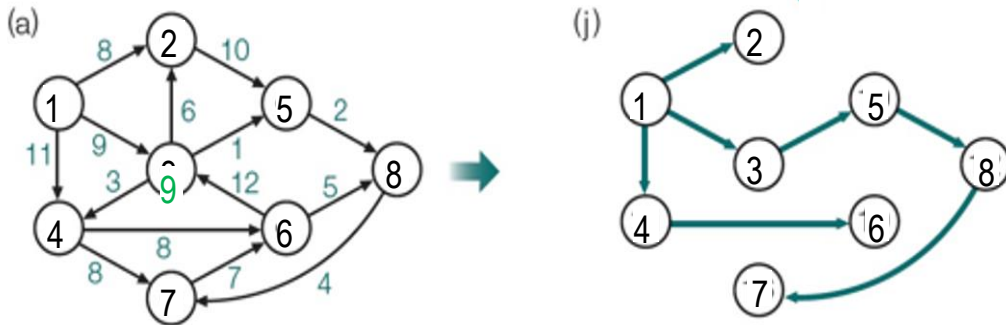
1 - 4

제시 문제

◆ [문제 2] 최단 경로 알고리즘 구현 [1/4]

– 제시된 `shortest_path.py`의 빈 라인을 채우세요

• Dijkstra 알고리즘



[주어지는 그래프]

```
dijkstra_adjacency_list = {  
    '1': {'2': 8, '3': 9, '4': 11},  
    '2': {'5': 10},  
    '3': {'2': 6, '5': 2, '4': 3},  
    '4': {'6': 9, '7': 8},  
    '5': {'8': 2},  
    '6': {'3': 12, '8': 5},  
    '7': {'6': 7},  
    '8': {'7': 4}  
}
```

주어진 코드 수정 불가

– 주어진 변수 이름 등도 수정 없이 그대로 사용

다음 페이지의 [숙제 코드 수행결과]와 동일한 수행 결과 출력 필요

[숙제 코드 수행결과]

[Graph] number of nodes: 8, number of edges: 14

{('5', '8', 2), ('1', '2', 8), ('6', '3', 12), ('7', '6', 7), ('1', '3', 9), ('3', '4', 3), ('3', '2', 6), ('2', '5', 10), ('8', '7', 4), ('4', '6', 9), ('6', '8', 5), ('1', '4', 11), ('3', '5', 2), ('4', '7', 8)}

*** Dijkstra Solution ***

1 → 2

1 → 3

1 → 4

3 → 5

4 → 6

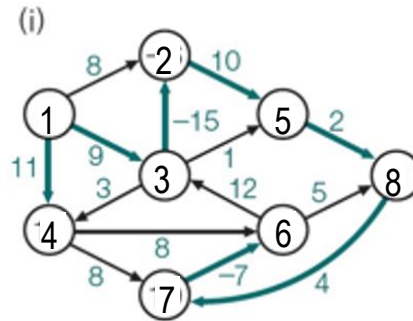
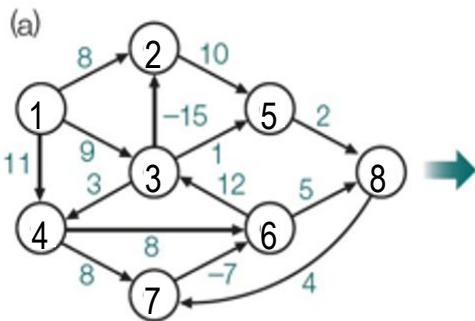
8 → 7

5 → 8

제시 문제

◆ [문제 2] 최단 경로 알고리즘 구현 (2/4)

- 제시된 [shortest_path.py](#)의 빈 라인을 채우세요
 - Bellman-ford 알고리즘



[주어지는 그래프]

```
bellman_ford_adjacency_list = {  
    '1': {'2': 8, '3': 9, '4': 11},  
    '2': {'5': 10},  
    '3': {'2': -15, '5': 1, '4': 3},  
    '4': {'6': 9, '7': 8},  
    '5': {'8': 2},  
    '6': {'3': 12, '8': 5},  
    '7': {'6': -7},  
    '8': {'7': 4}  
}
```

[숙제 코드 수행결과]

[Graph] number of nodes: 8, number of edges: 14

{('5', '8', 2), ('1', '2', 8), ('3', '5', 1), ('6', '3', 12), ('1', '3', 9), ('4', '7', 8), ('3', '4', 3), ('2', '5', 10), ('7', '6', -7), ('4', '6', 9), ('6', '8', 5), ('1', '4', 11), ('8', '7', 4), ('3', '2', -15)}

*** Bellman Ford Solution ***

3 → 2

1 → 3

1 → 4

2 → 5

7 → 6

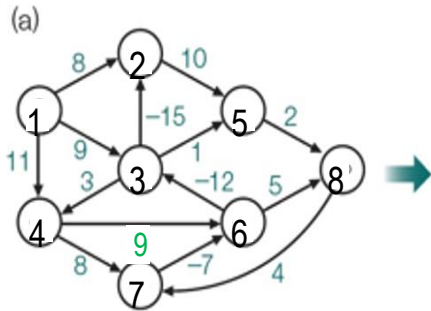
5 → 8

8 → 7

제시 문제

◆ [문제 2] 최단 경로 알고리즘 구현 (3/4)

- 제시된 `shortest_path.py`의 빈 라인을 채우세요
 - Bellman-ford 알고리즘: 주어진 그래프에 음의 사이클이 있는 경우



[주어지는 그래프]

```
bellman_ford_adjacency_list_with_negative_cycle = {
```

```
'1': {'2': 8, '3': 9, '4': 11},
'2': {'5': 10},
'3': {'2': -15, '5': 1, '4': 3},
'4': {'6': 9, '7': 8},
'5': {'8': 2},
'6': {'3': -12, '8': 5},
'7': {'6': -7},
'8': {'7': 4}
```

```
}
```

LINK@KOREATEC

[숙제 코드 수행결과]

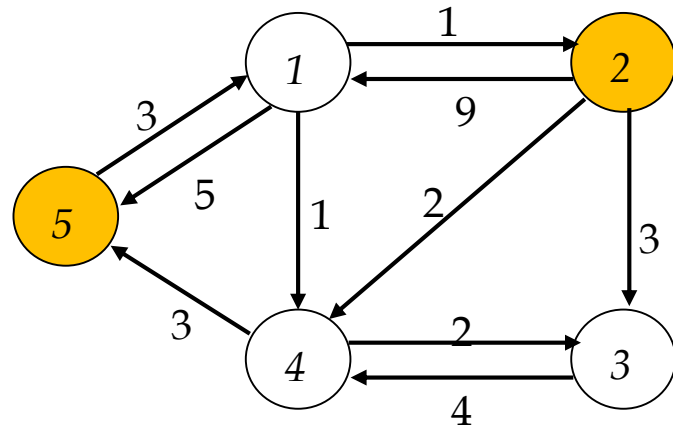
```
[Graph] number of nodes: 8, number
of edges: 14
{('5', '8', 2), ('1', '2', 8), ('3', '5', 1), ('1', '3',
9), ('4', '7', 8), ('3', '4', 3), ('2', '5', 10), ('6',
'3', -12), ('7', '6', -7), ('4', '6', 9), ('6', '8',
5), ('1', '4', 11), ('8', '7', 4), ('3', '2', -15)}
```

음의 사이클 발견! 해 없음

제시 문제

◆ [문제 2] 최단 경로 알고리즘 구현 (4/4)

- 제시된 `shortest_path.py`의 빈 라인을 채우세요
 - floyd-warshall 알고리즘



[주어지는 그래프]

```
floyd_warshall_adjacency_list = {  
    '1': {'2': 1, '4': 1, '5': 5},  
    '2': {'1': 9, '3': 3, '4': 2},  
    '3': {'4': 4},  
    '4': {'3': 2, '5': 3},  
    '5': {'1': 3}  
}
```

[숙제 코드 수행결과]

```
[Graph] number of nodes: 5, number  
of edges: 10  
{('3', '4', 4), ('2', '1', 9), ('2', '3', 3), ('1', '2',  
1), ('4', '3', 2), ('1', '5', 5), ('4', '5', 3), ('5',  
'1', 3), ('1', '4', 1), ('2', '4', 2)}
```

2 4 5 1
5 1 4 3
2 4

제시 문제

◆ [문제 3]

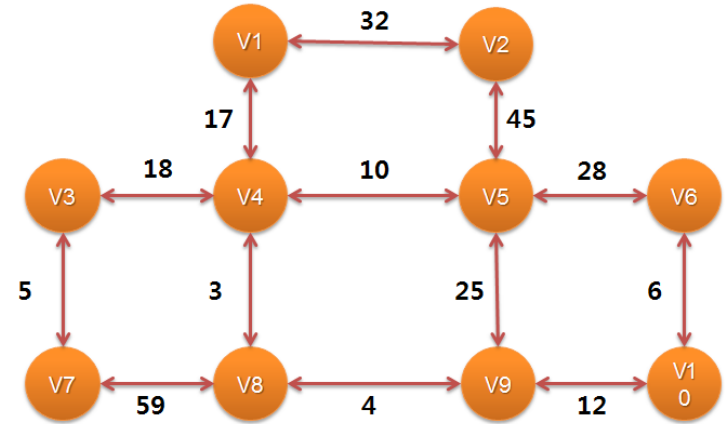
- 연쇄 행렬들의 곱인 $A_1 \times A_2 \times A_3 \times A_4 \times A_5$ 를 계산하는 최적 순서와 그 비용을 구하라.
 - 주어진 행렬들의 크기는 다음과 같다.

$$\begin{array}{ccccccccc} A_1 & \times & A_2 & \times & A_3 & \times & A_4 & \times & A_5 \\ 10 \times 4 & & 4 \times 5 & & 5 \times 20 & & 20 \times 2 & & 2 \times 50 \end{array}$$

- [참고 1] 본 문제는 수업시간에 배운 대로 위 문제에 대한 재귀적 속성을 올바르게 이해하고 그에 따른 Dynamic Programming 풀이 절차를 이해하는 지에 대한 자기 스스로의 검증차원의 문제임
- [참고 2] 다소 풀이가 길어질 수 있으므로 한글이나 워드 리포트에 해답을 직접 타이핑하는 것 보다 종이 연습장에 깔끔하게 손으로 풀이를 하고, 사진을 선명하게 찍어 해당 사진 이미지를 보고서에 붙이는 식으로 구성을 하면 좋을 것임

제시 문제

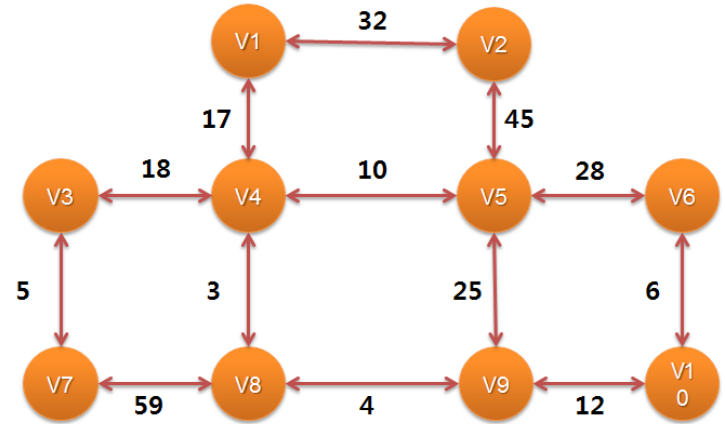
◆ [문제 4]



- [문제 4-1] Prim 알고리즘을 이용하여 위 그래프의 최소비용 신장 트리를 구하는 과정을 교재 [그림 10-16]과 유사한 형태로 그려 제시하시오.
 - 캡처 이미지 제시
- [문제 4-2] Kruskal 알고리즘을 이용하여 위 그래프의 최소비용 신장 트리를 구하는 과정을 교재 [그림 10-17]과 유사한 형태로 그려 제시하시오.
 - 캡처 이미지 제시
- [문제 4-3] 앞선 [문제 1]에서 작성한 최소 비용 신장 트리 알고리즘 파이썬 코드를 이용하여 위 [4-1] 및 [4-2]에 제시한 해답을 검증하시오.
 - 캡처 이미지 제시

제시 문제

◆ [문제 5]

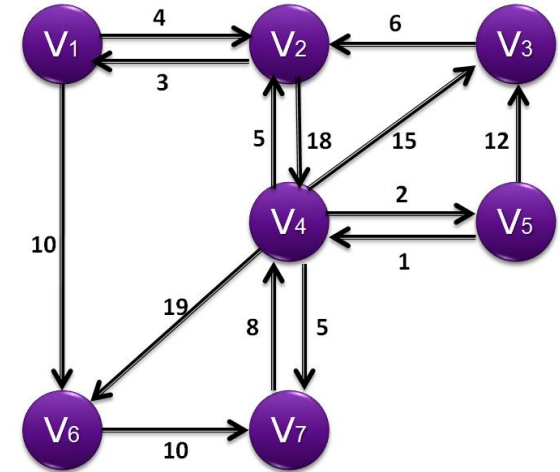


- [문제 5-1] Dijkstra 알고리즘을 이용하여 위 그래프[앞선 [문제 4와 동일한 그래프]에서 정점 v_4 에서 다른 모든 정점으로 가는 최단경로를 구하는 과정을 교재 [그림 10-23] 과 유사한 형태로 그려 제시하시오. 여기서 각 무향 간선은 같은 가중치를 가진 2개의 쌍방향 간선을 나타낸다고 가정하시오.
 - 캡처 이미지 제시
- [문제 5-2] 앞선 [문제 2]에서 작성한 Dijkstra 알고리즘 파이썬 코드를 이용하여 위 [5-1] 에 제시한 해답을 검증하시오.
 - 캡처 이미지 제시

제시 문제

◆ [문제 6]

- [6-1] 오른쪽 주어진 그래프에서 Floyd-Warshall 알고리즘에서 d_{ij}^k 와 p_{ij}^k 가 만들어지는 과정을 그림으로 제시하시오.
 - d_{ij}^k 와 p_{ij}^k 각각 2차원 행렬로 간주하고 총 7 단계별로 2차원 행렬 각각 1개씩 차례로 제시하시오.
- [6-2] [3-1]에서 제시한 p_{ij}^k 의 마지막 행렬을 이용하여 다음 2가지 경우에 대한 경로를 풀이과정과 함께 제시하시오
 - Case1] 출발: v_1 , 도착: v_5
 - Case2] 출발: v_3 , 도착: v_6
- [6-3] 앞선 [문제 2]에서 작성한 Floyd-Warshall 알고리즘 파이썬 코드를 이용하여 위 [6-2]에 제시한 해답을 검증하시오.
 - 캡처 이미지 제시



제시 문제

◆ [문제 7]

- [문제 7-1] 허프만의 알고리즘을 사용하여 다음 표에 있는 글자들에 대한 최적 이진 트리를 구축하는 과정을 제시하시오.
 - 캡처 이미지 제시

글자	A	B	I	M	S	X	Z
빈도수	12	7	18	10	9	5	2

- [문제 7-2] 위 [7-1]에서 제시한 최적 이진 트리를 기반으로 각 주어진 문자에 대한 최적 전치 코드(허프만 코드)를 제시하시오.

문자	허프만 코드
A	
B	
I	
M	
S	
X	
Z	

숙제 요구사항

◆ 보고서(Report) 요구사항

- 프로그램 코드를 정확하게 Report에 작성할 것
- 주석을 충분하게 넣을 것
- 프로그램 내에 작성한 특정한 부분 코드에 대한 설명이 구체적으로 필요한 경우는 프로그램 전체 코드 바로 다음에 해당 설명을 적을 것
- 프로그램 수행 화면을 캡처하여 넣을 것
- 캡처화면은 다양하게 많이 넣을 것
- 자신이 취득한 기술적 사항/고찰 내용을 생각한 바 그대로 적을 것

숙제 요구사항

◆ 제출형태 및 방법

– 보고서 파일 작성

- HWP 또는 MS-Word로 작성하여 **PDF 산출 및 제출**
- 서론, 본론, 결론의 구조를 가진 보고서
 - 표지
 - 서론: Homework의 내용 및 목적
 - 본론: 각각의 문제를 코딩한 이후 앞에 있는 요구사항에 따라 작성
 - 결론: 숙제를 한 이후의 느낀점, 하고싶은 말, 또는 불평~~
- 파일명: 알고리즘-4차-홍길동-2020111222.pdf

– 숙제 제출 방식

- <http://el.koreatech.ac.kr> 의 “과제 ” 메뉴
- 숙제 게시물 제목: 알고리즘-4차-홍길동-2020111222

– 숙제를 포함한 여러가지 질문 방법

- <http://el.koreatech.ac.kr> 의 “질의응답” 게시판 활용할 것