

# Programming Assignment #2 - Part 2

학번 / 학과 : B889047 / 소프트웨어 융합학과

작성자 : 윤준호

## Part 2

Topic : Implement Prim's MST(Minimum Spanning Tree)알고리즘을 구현한다

< Prim's Pseudo Code / Format: Scala language >

```
Prim's MST algorithm --
  Q ← V
  key[v] ← ∞
  key[s] ← 0
  while Q ≠ ∅
    do u ← EXTRACT-MIN(Q)
    for each v ∈ Adj[u] // Adjacency list
      do if (v ∈ Q and w(u,v) > key[v] )
        then key[v] ← w(u, v) // Decrease-Key operation
        p[v] ← u
```

### Install dependency packages from pip

In [1]:

```
!pip3 install -r requirements.txt
```

```
Requirement already satisfied: treelib in /Users/hoplin/opt/anaconda3/
lib/python3.9/site-packages (from -r requirements.txt (line 1)) (1.6.
1)
Requirement already satisfied: networkx in /Users/hoplin/opt/anaconda
3/lib/python3.9/site-packages (from -r requirements.txt (line 2)) (2.
7.1)
Requirement already satisfied: future in /Users/hoplin/opt/anaconda3/l
ib/python3.9/site-packages (from treelib->-r requirements.txt (line
1)) (0.18.2)
```

### 들어가기 앞서

결론적으로 프림알고리즘을 Part1의 우선순위 큐로 구현하는데 실패하였습니다.제가 작성한 우선순위 큐는 Object 혹은 Key-Value타입에 대한 것을 지원하지 못하는 우선순위 큐입니다.

그렇기에 Value와 Key값을 병렬적으로 처리하려 하였으나, 이 방법에도 실패하였습니다. 조금 더 깊은 생각과 우선순위 큐의 더 넓은 범위의 범용성을 생각하여 재설계가 필요하다고 느꼈습니다. 해당 부분에 있어서는 학기 이후에 더 개선해 보도록 하겠습니다. 이로 인해 문제 A는 해결하지 못하였음을 알려드립니다

## 아래 두 문제를 한번에 해결하는 과정을 보이겠습니다

### 문제 B

Q : 위 알고리즘에는 경로 정보가 저장된다. MST를 찾은 경우 저장된 경로를 출력하는 프로그램을 작성해라

### 문제 C

Q : 테스트 케이스를 하나 만들어 결과를 출력해 보인다.

## 테스트로 사용할 인접행렬 형태의 그래프와 Prim Algorithm 인스턴스 생성하기

- 작성한 프림 알고리즘 클래스의 생성자는 두개의 매개변수를 받습니다
  - 그래프
    - `change_graph(graph_matrix)` 메소드로 그래프를 변경할 수 있습니다.
  - 시작정점
    - `change_starting_node(node)` 메소드로 시작정점을 변경할 수 있습니다.

In [2]:

```
from math import inf
from PrimAlgorithm.PrimAlgorithm import PrimAlgorithm
```

In [3]:

```
# 그래프
graph = [
    [0, 2, 5, 1, inf, inf],
    [2, 0, 3, 2, inf, inf],
    [5, 3, 0, 3, 1, 5],
    [1, 2, 3, 0, 1, inf],
    [inf, inf, 1, 1, 0, 2],
    [inf, inf, 5, inf, 2, 0]
]
# 시작노드
starting_point = 0

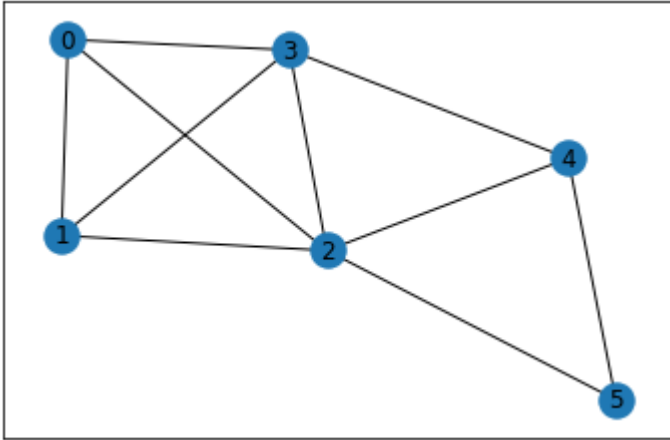
prim = PrimAlgorithm(graph, starting_point)
```

## 테스트 케이스로 초기화한 그래프 Visualization

테스트 케이스로 사용할 그래프를 시각화해 봅니다. 시각화하기 위해서는 PrimAlgorithm인스턴스의 `visualizeInitialGraph()` 메소드를 사용하면 됩니다

In [4]:

```
prim.visualizeInitialGraph()
```



## Prim MST Algorithm 실행 및 Summary 출력

- PrimAlgorithm인스턴스의 prim()메소드를 실행하면 됩니다. prim()메소드는 프림 알고리즘 실행 후에 Summary 정보를 출력합니다.
- Summary 정보는 아래와 같은 정보들을 출력합니다
  - MST를 이루는 간선들과 그 간선의 가중치
  - MST Visualization

In [5]:

```
prim.prim()
```

[ Prim MST's edge information ]

```
0 -> 3 | Weight : 1  
3 -> 4 | Weight : 1  
4 -> 2 | Weight : 1  
0 -> 1 | Weight : 2  
4 -> 5 | Weight : 2
```

[ Prim MST's Visualization ]

