

# アルゴリズム論

2017年7月3日

樋口文人

# 目次

- グラフ

- 問題

- トポロジカルソート
    - 最小全域木
    - 最短経路
    - 巡回セールスマン問題

DAGの

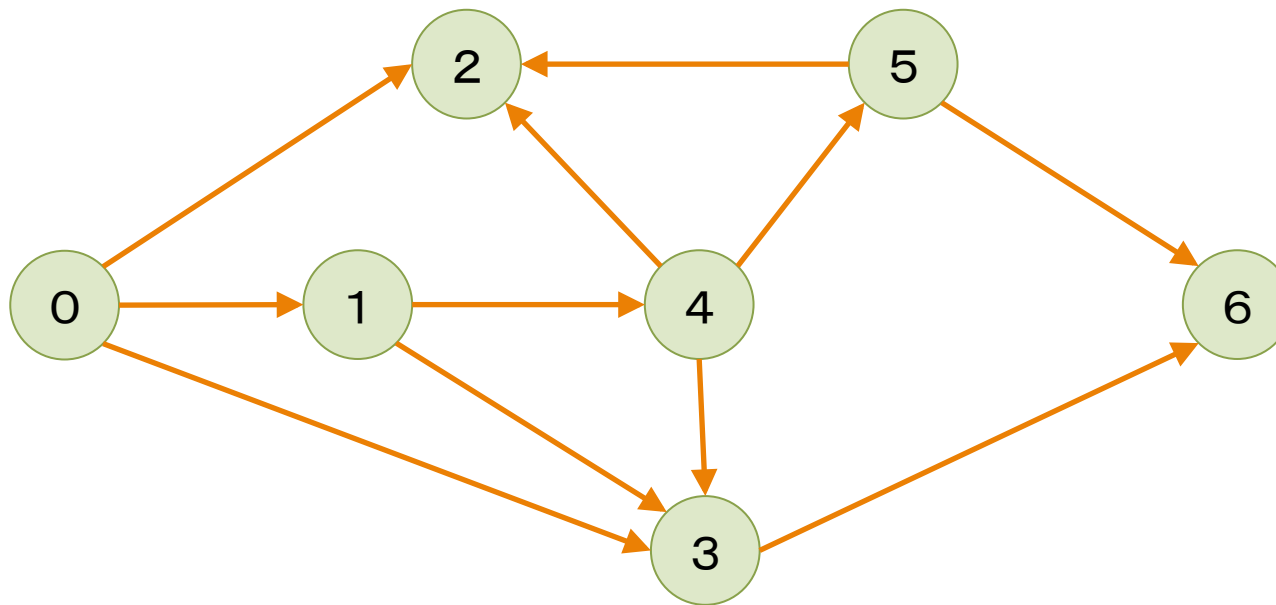
# トポロジカルソート

# 問題

- 履修科目の前提要件を満たす履修方法
- シンボリックリンクのループ検出
- 表計算ソフトの計算式と計算順序
- 相互に依存するプログラムのコンパイル順序
- PERT: Project Evaluation and Review Technique

# DAG: Directed Acyclic Graph

循環路の無い有向グラフ

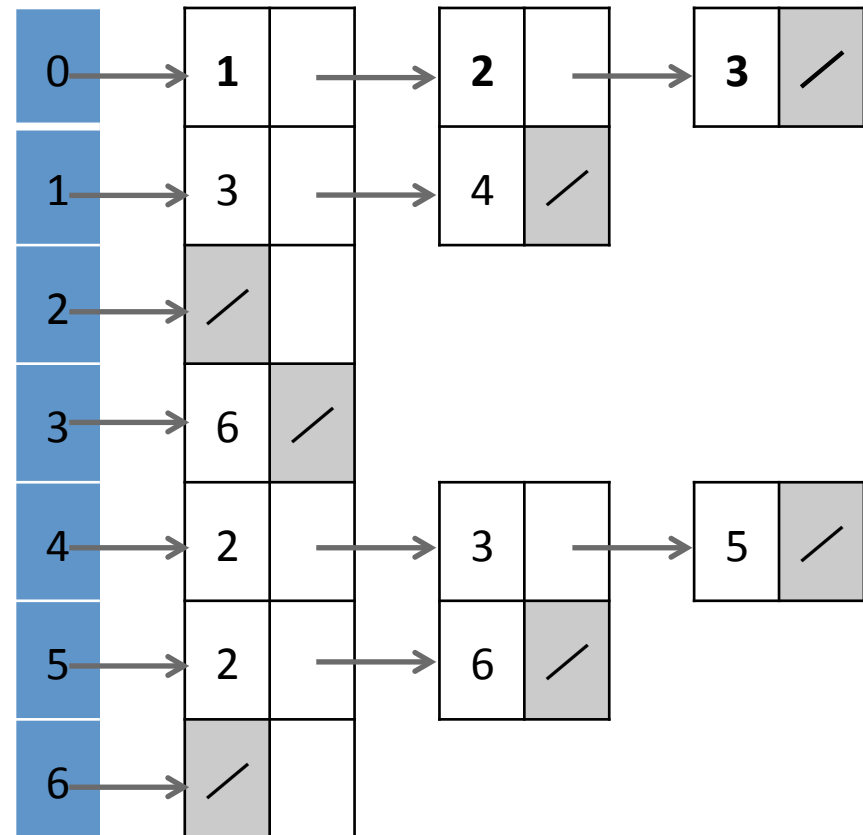


# グラフの表現：どこに隣接しているか？

## 隣接行列

	0	1	2	3	4	5	6
0	0	1	1	1	0	0	0
1	0	0	0	1	1	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1
4	0	0	1	1	0	1	0
5	0	0	1	0	0	0	1
6	0	0	0	0	0	0	0

## 隣接リスト

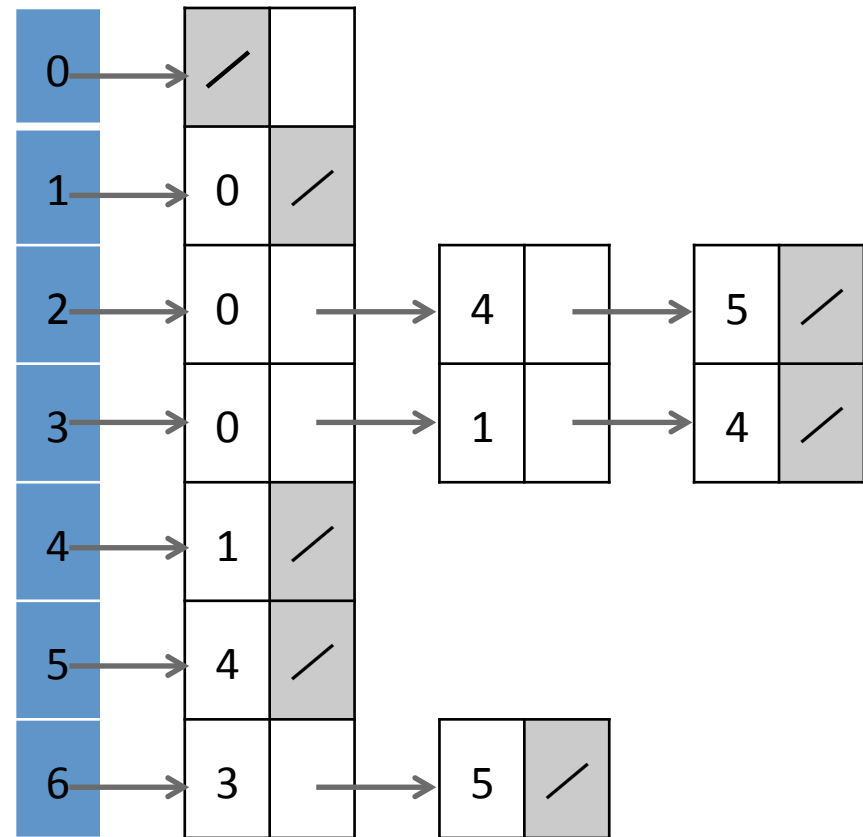


# グラフの表現：どこから隣接しているか？

## 隣接行列の転置行列

	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
2	1	0	0	0	1	1	0
3	1	1	0	0	1	0	0
4	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0
6	0	0	0	1	0	1	0

## 隣接リスト

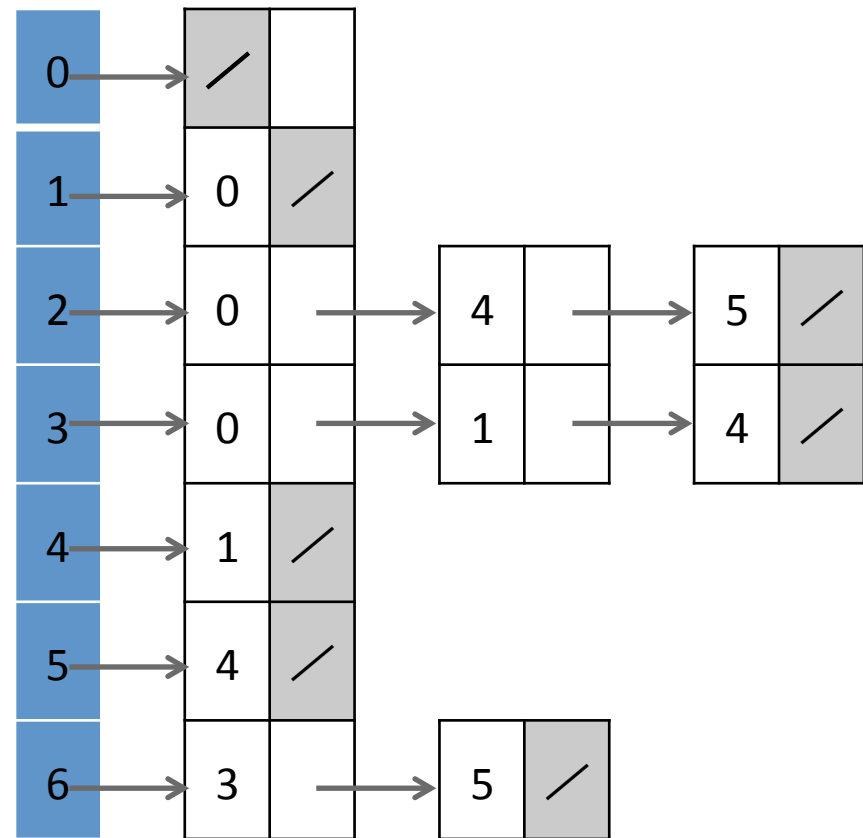


# トポロジカルソート 方法

キュー: 0

- どこからも入ってくる辺が無い  
節点0をキューに追加
  - 節点の順番を示す番号を0に設定
- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト



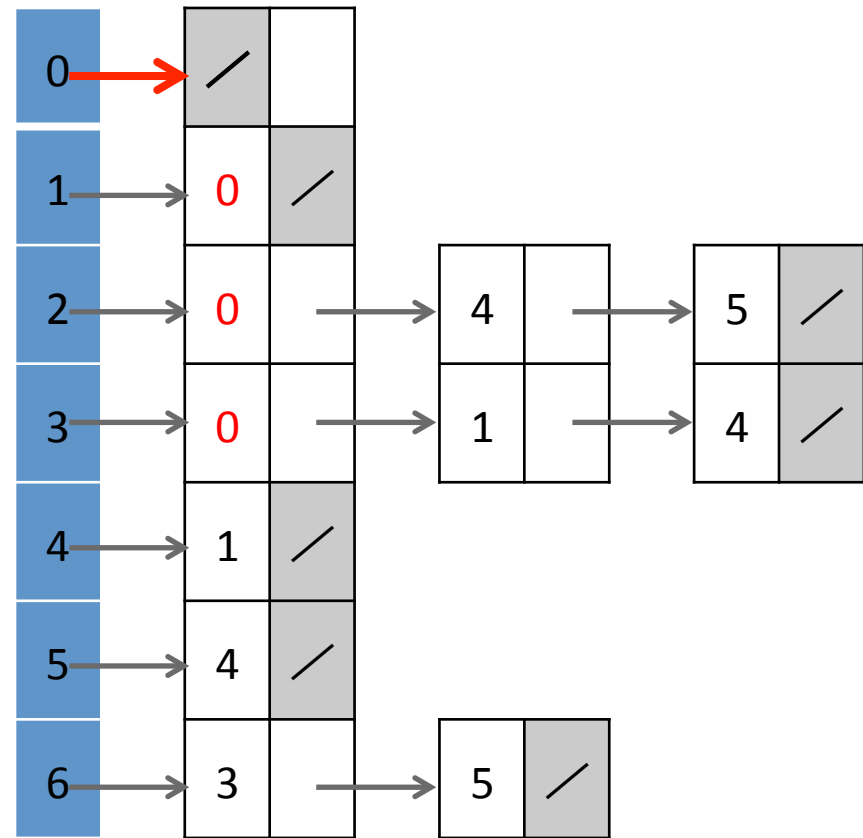


# トポロジカルソート step 0

キュー: 0

- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える(0)
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト

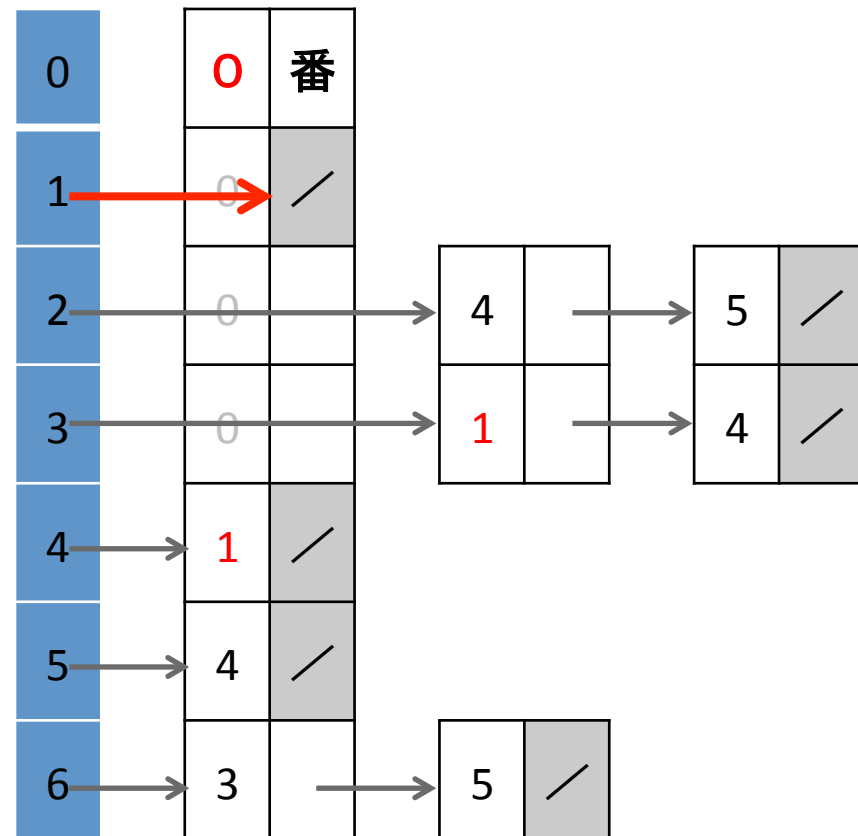


# トポロジカルソート step 1

キュー: **1**

- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える(**1**)
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト

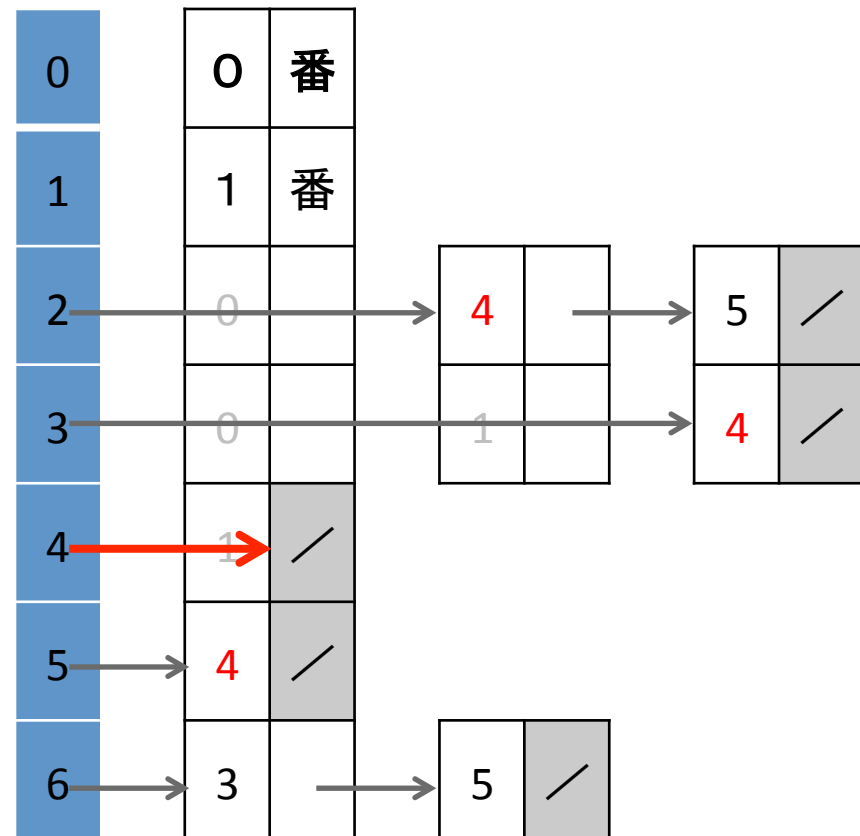


# トポロジカルソート step 2

キュー: 4

- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える(2)
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト

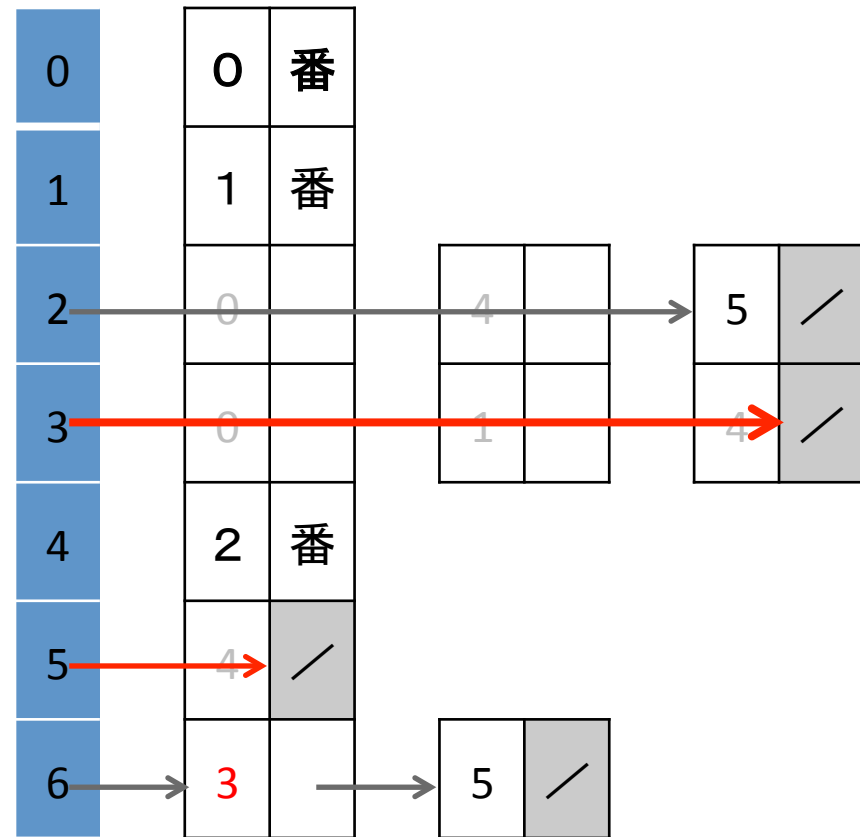


# トポロジカルソート step 3

キュー: 3, 5

- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える(3)
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト

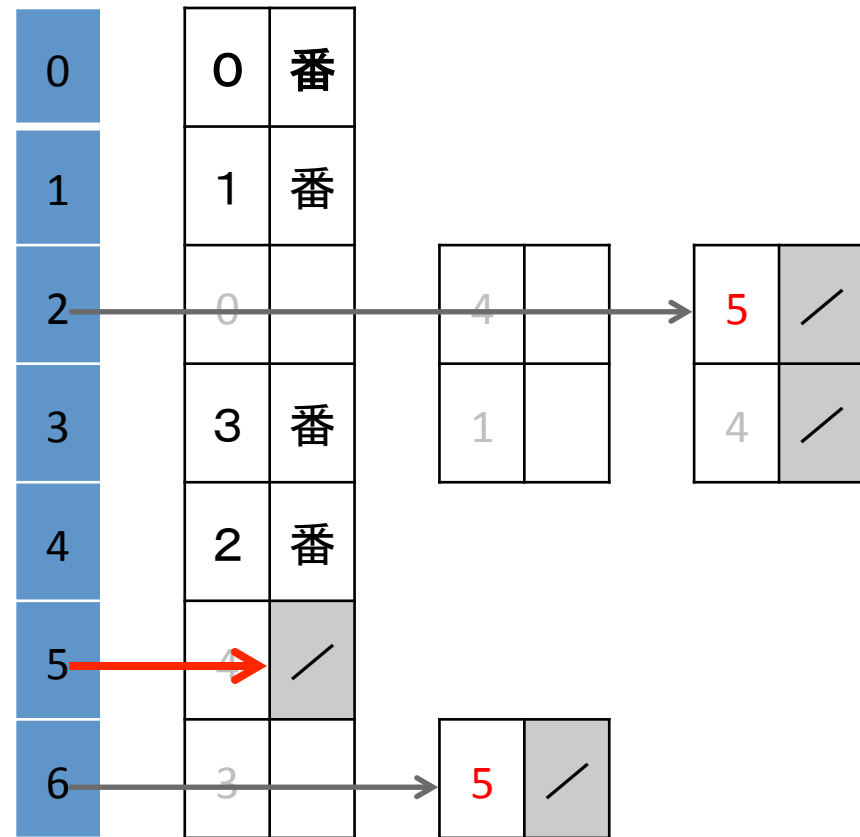


# トポロジカルソート step 4

キュー: 5

- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える(4)
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト

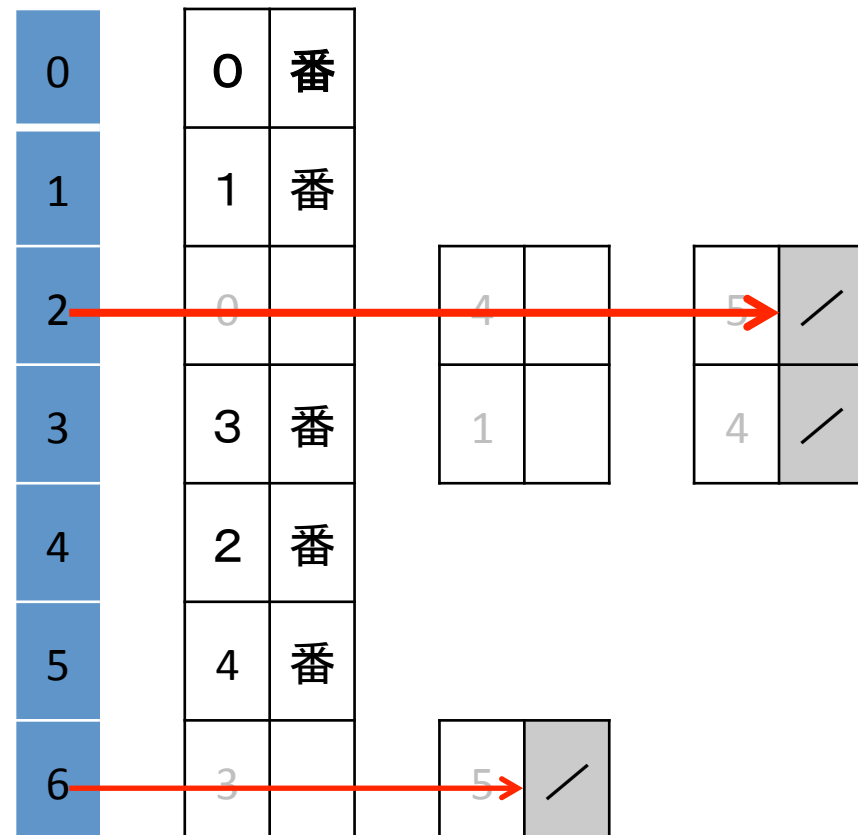


# トポロジカルソート step 5

キュー: 2, 6

- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える(5)
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト

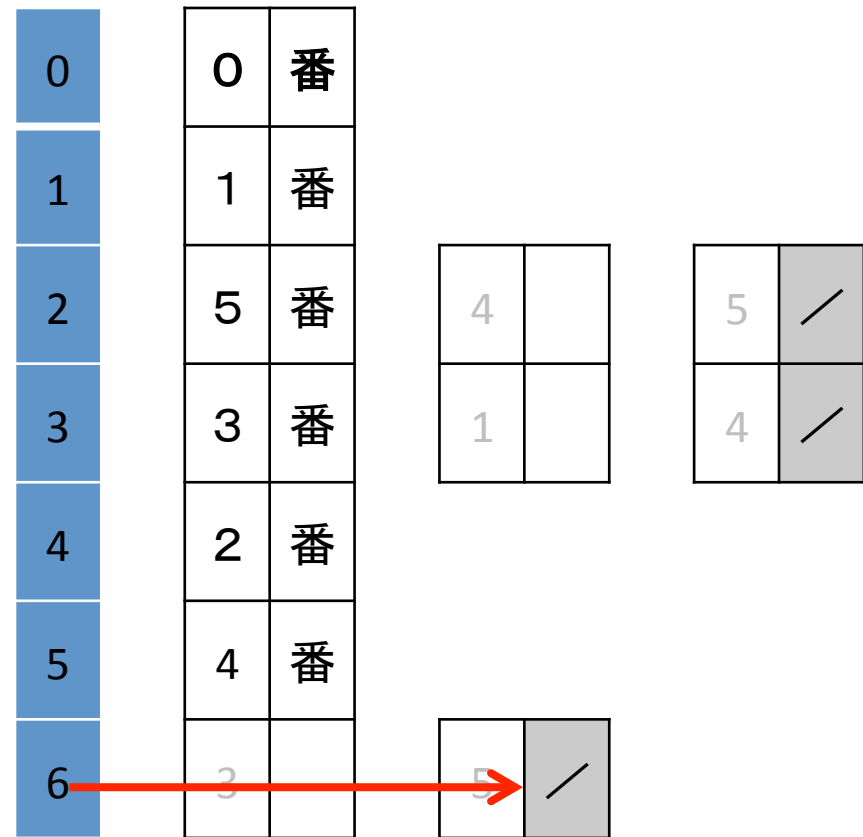


# トポロジカルソート step 6

キュー: **6**

- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える(**6**)
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト



# トポロジカルソート step 7

キュー:

- キューが空になるまで以下繰り返し
  - キューから節点を1つ取り出す
  - この節点に番号を与える(7)
  - 番号を1つ増やす
  - 隣接リストからこの節点を削除
  - 空のリストがあればその節点をキューに追加

隣接リスト

0	0	番		
1	1	番		
2	5	番	4	
3	3	番	1	
4	2	番		
5	4	番		
6	6	番	5	／

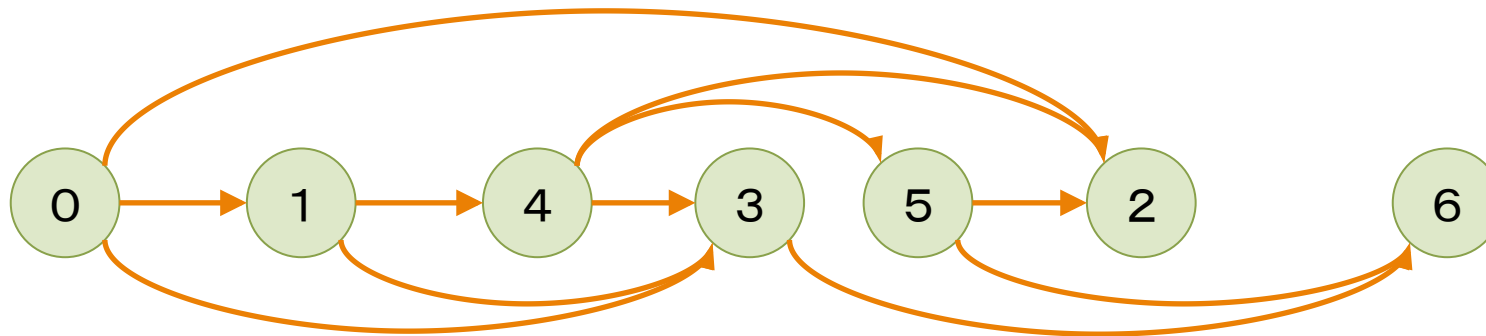

5	／
4	／

5	／
---	---



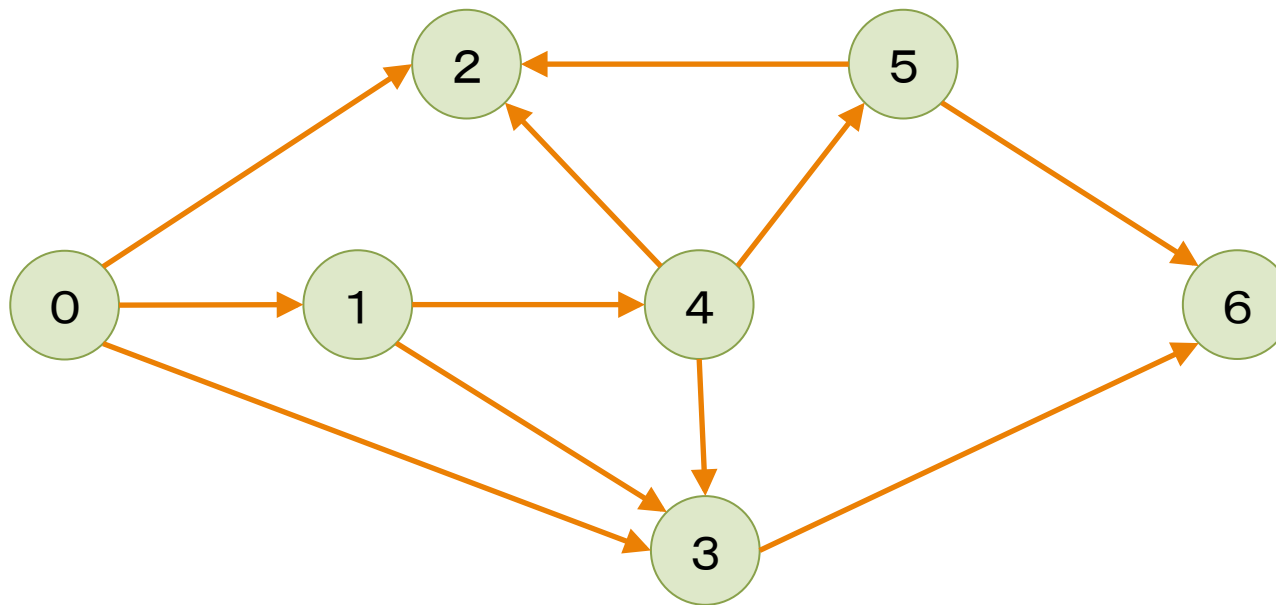
# トポロジカルソート 結果1

終了時の番号(7)と節点の数が一致: 閉路はない



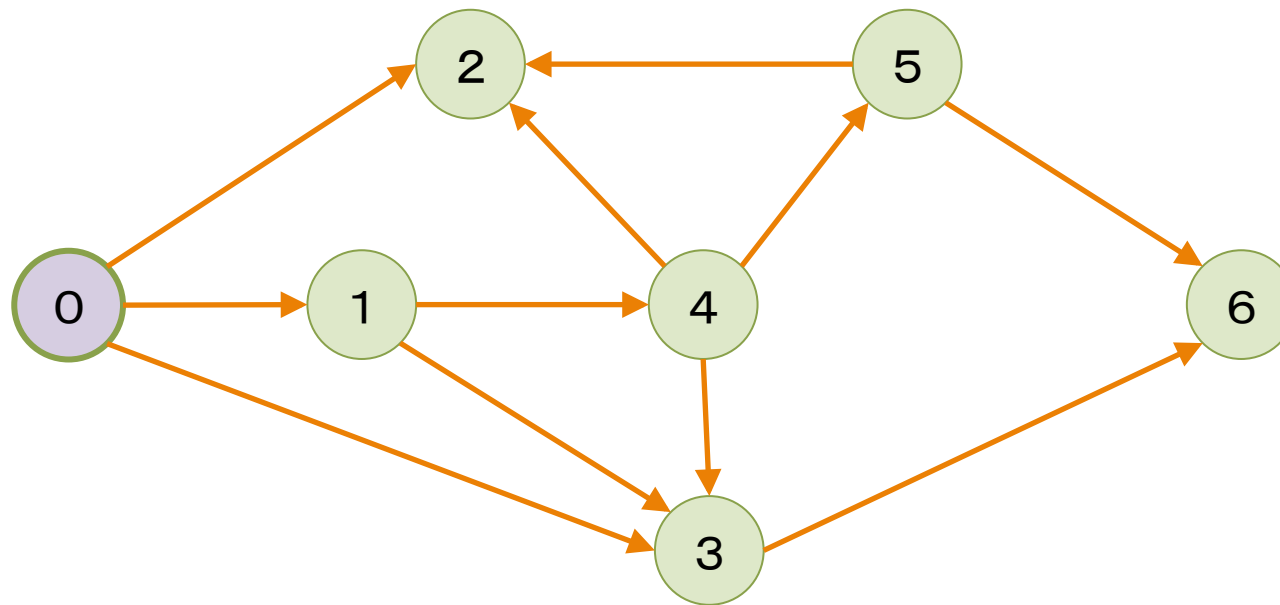
# トポロジカルソート 方法2

DFSの応用: バックトラックを含めた訪問順を記録する



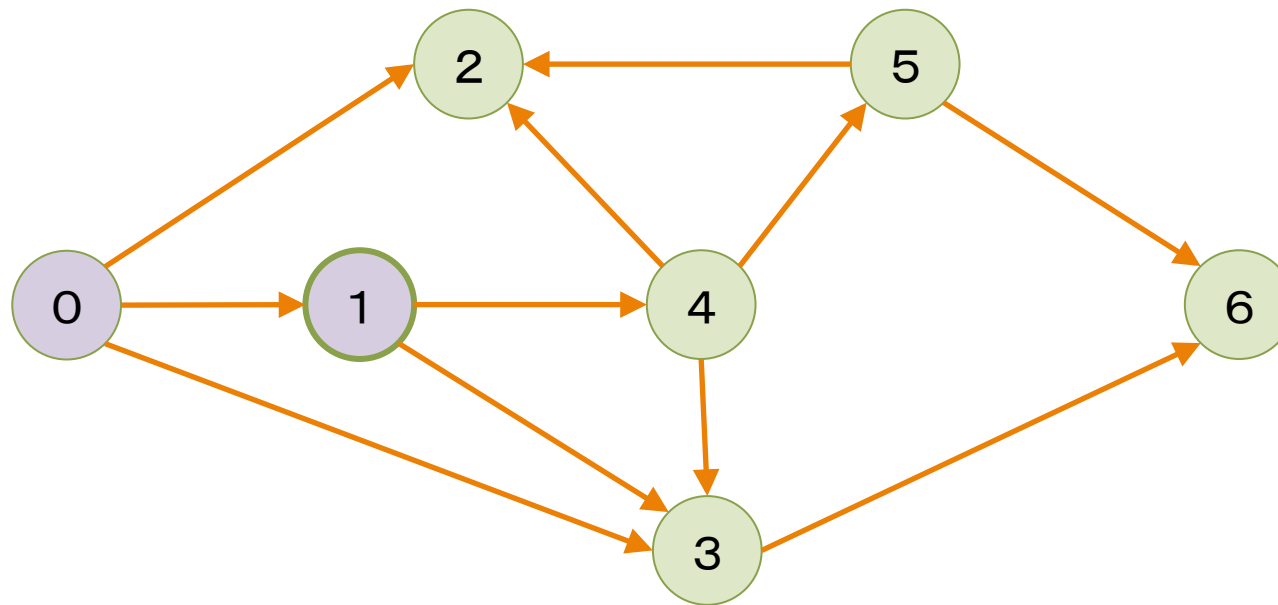
# トポロジカルソート 方法2

訪問順: 0



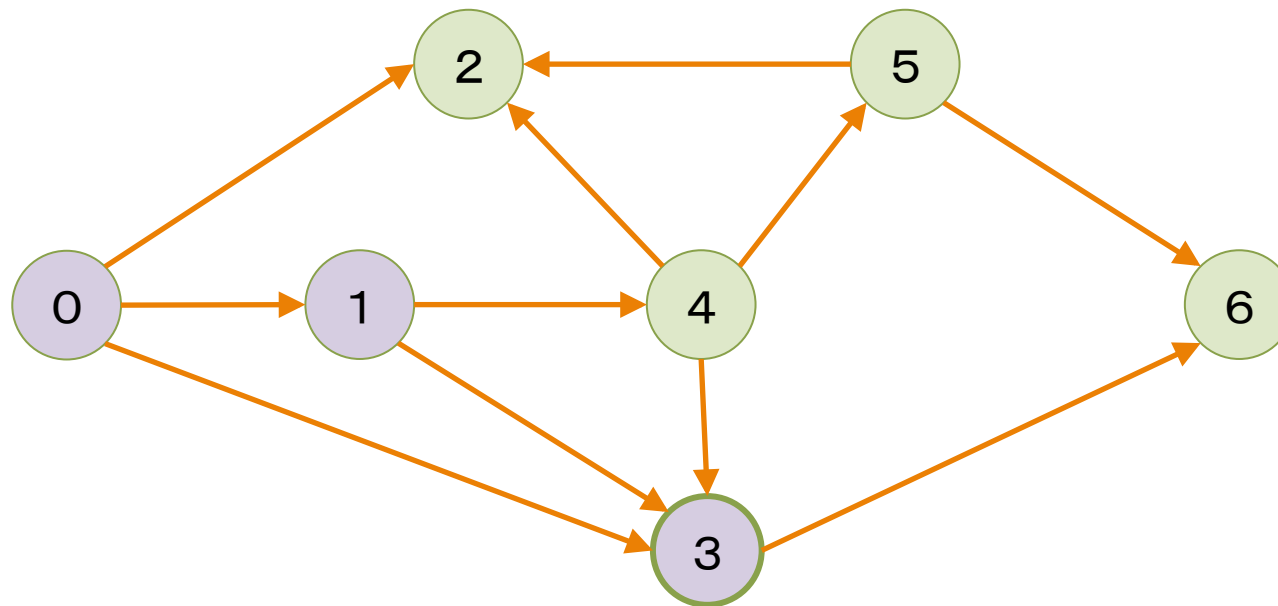
# トポロジカルソート 方法2

訪問順: 01



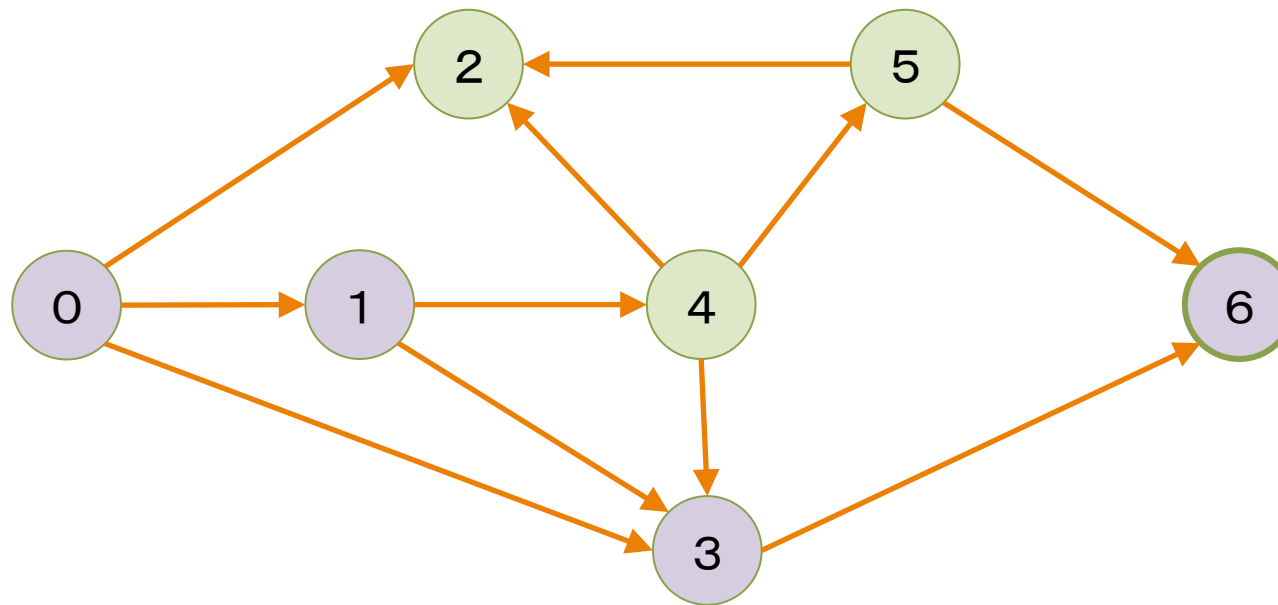
# トポロジカルソート 方法2

訪問順: 013



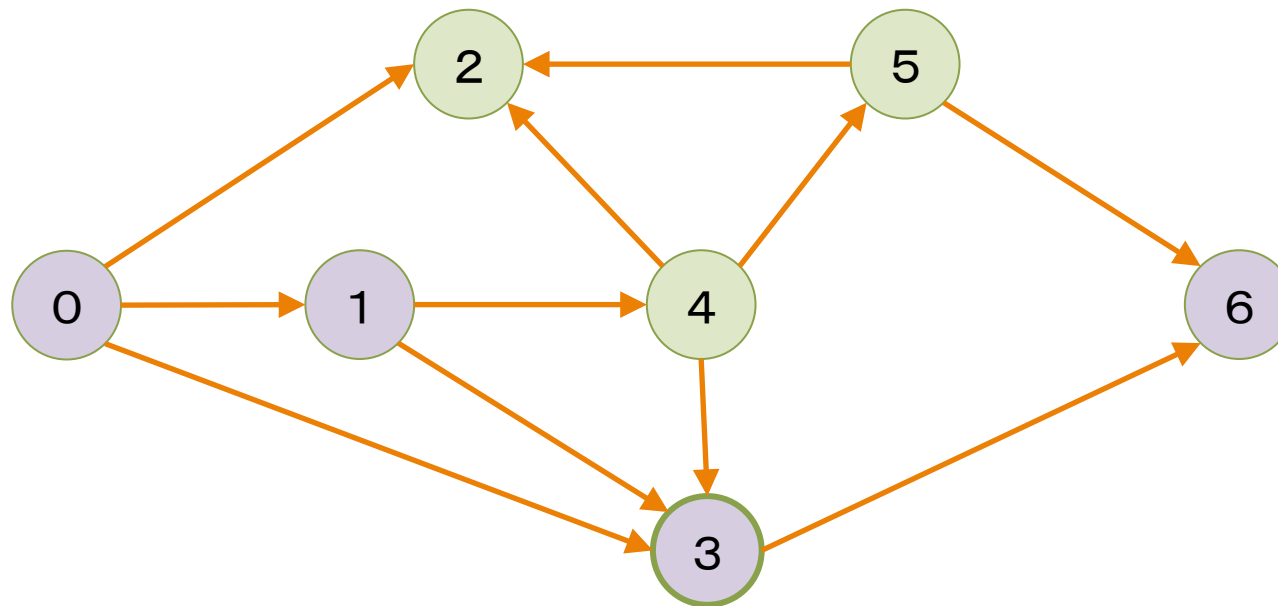
# トポロジカルソート 方法2

訪問順: 0136



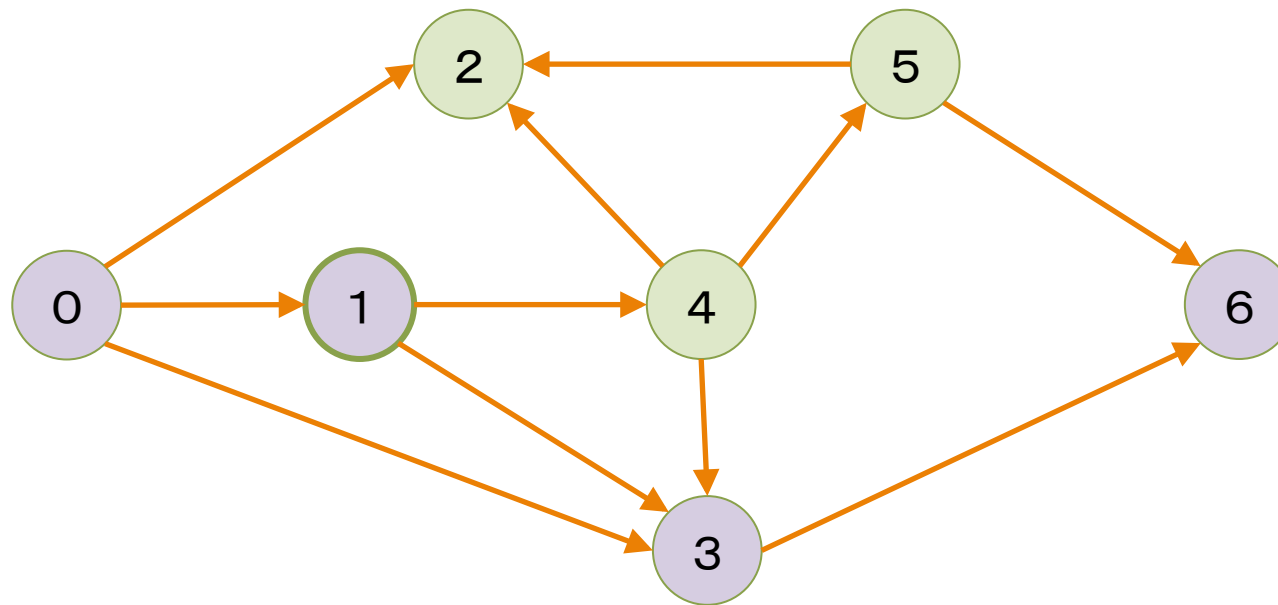
# トポロジカルソート 方法2

訪問順: 01363



# トポロジカルソート 方法2

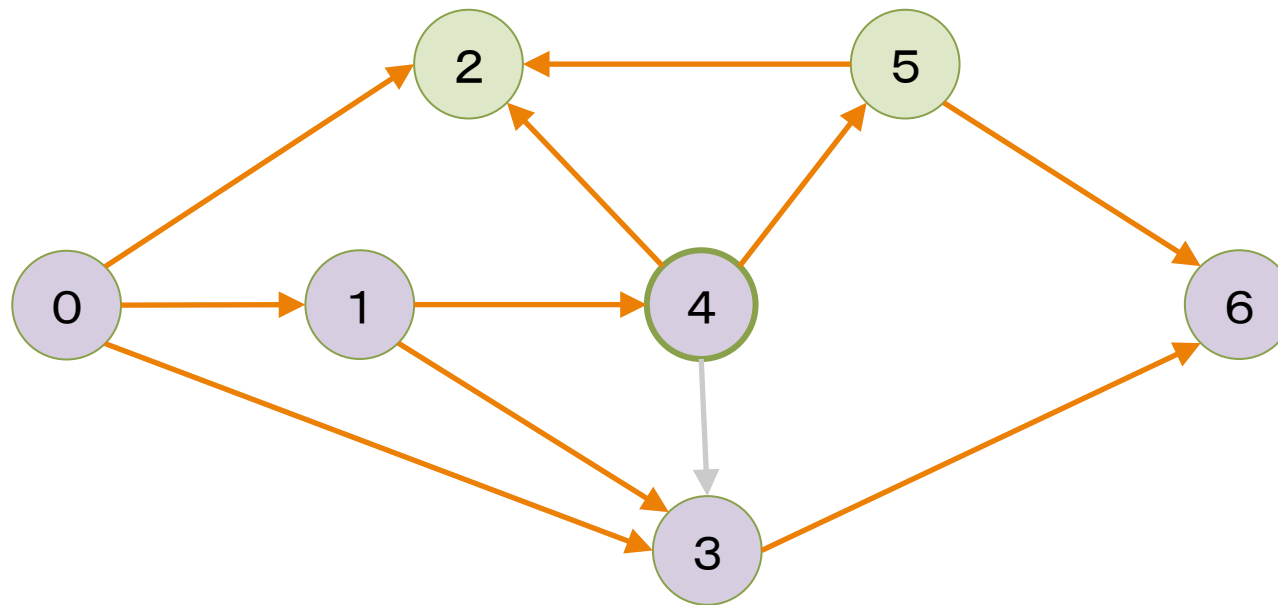
訪問順: 013631





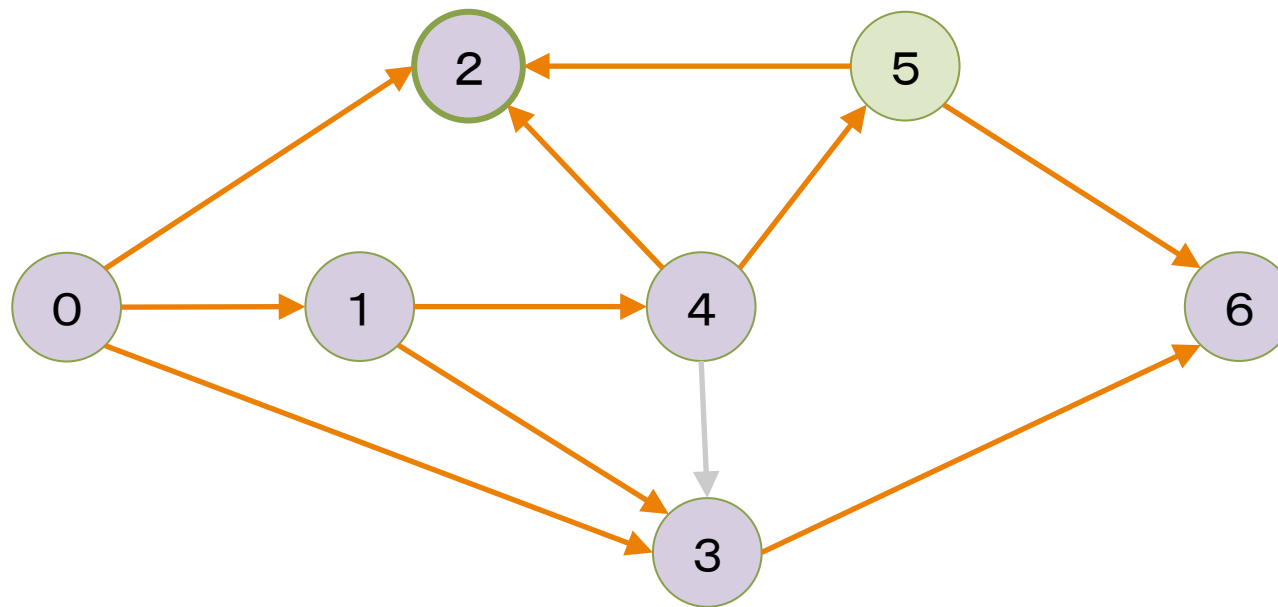
# トポロジカルソート 方法2

訪問順: 0136314



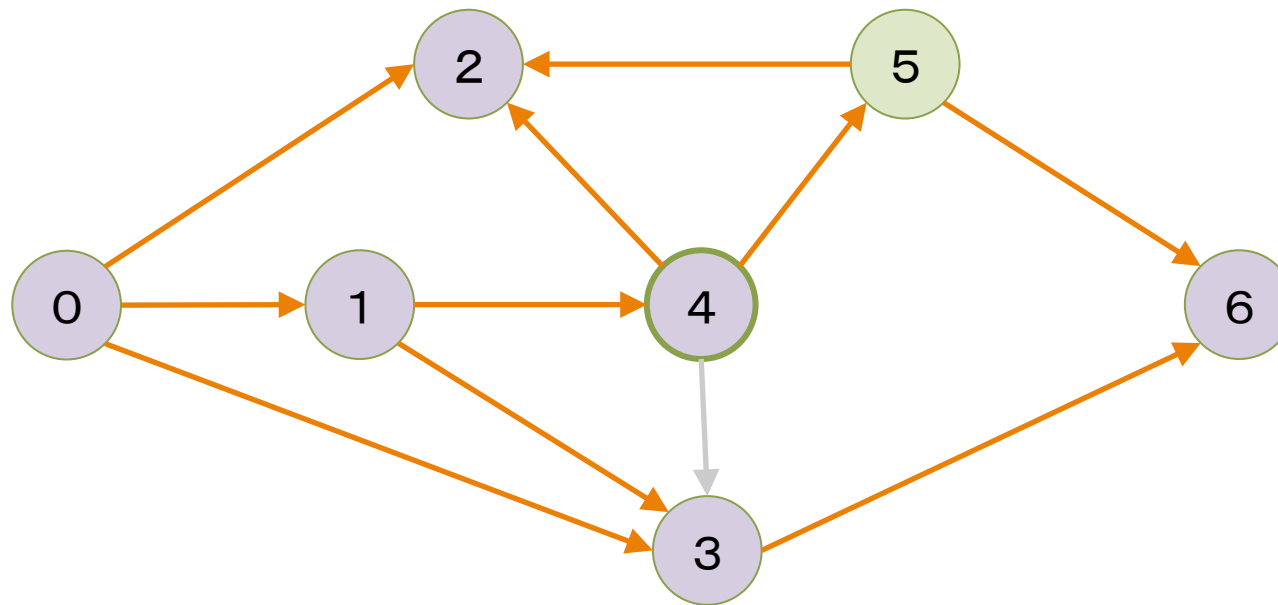
# トポロジカルソート 方法2

訪問順: 013~~6~~3142



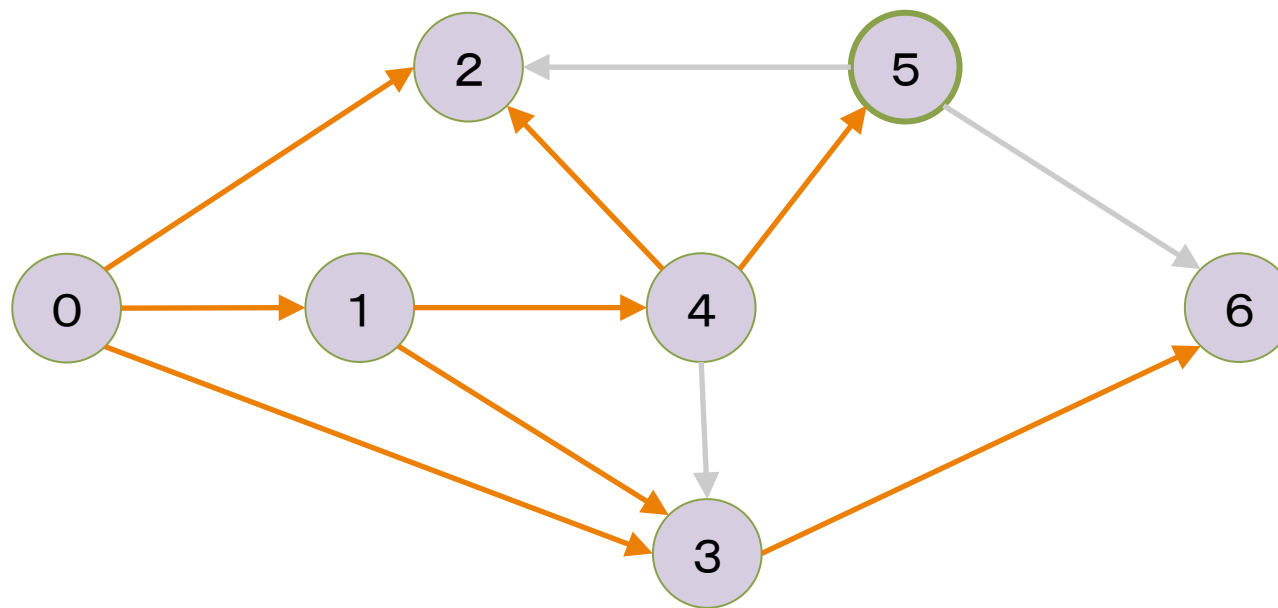
# トポロジカルソート 方法2

訪問順: 013631424



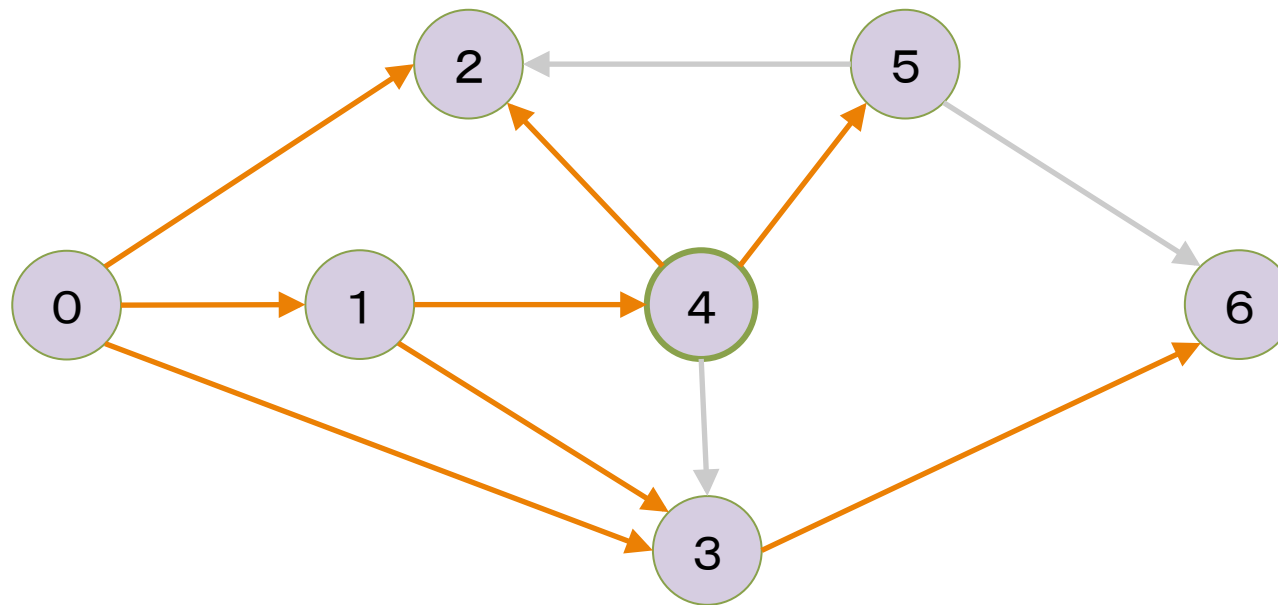
# トポロジカルソート 方法2

訪問順: 0 1 3 6 3 1 4 2 4 5



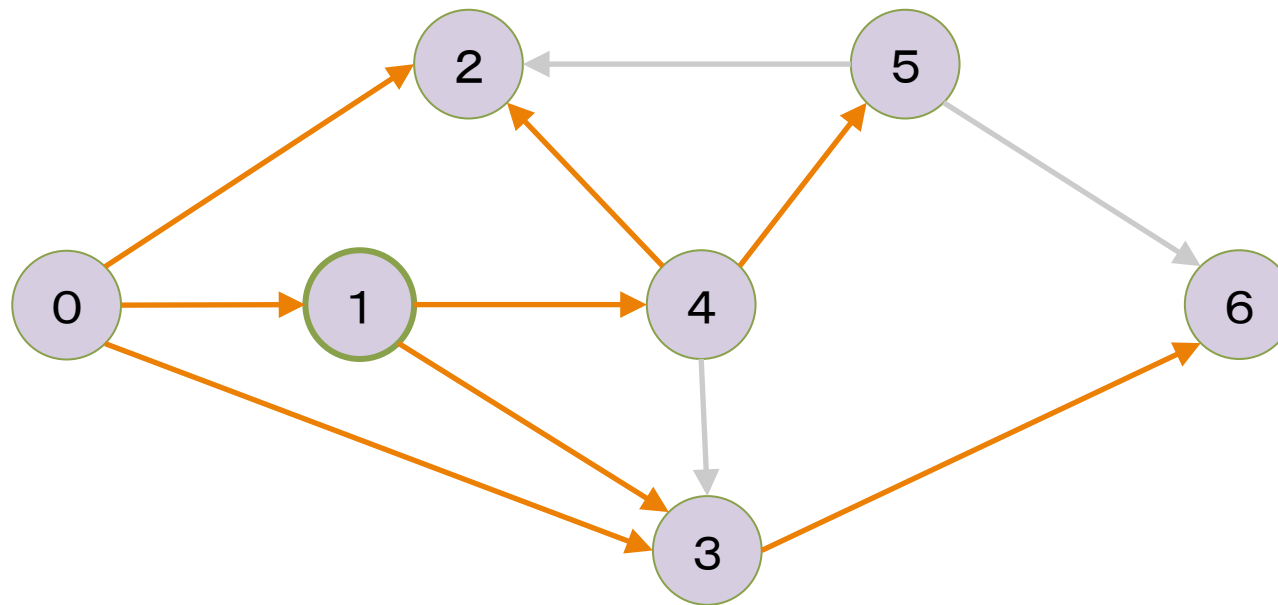
# トポロジカルソート 方法2

訪問順: 0 1 3 6 3 1 4 2 4 5 4



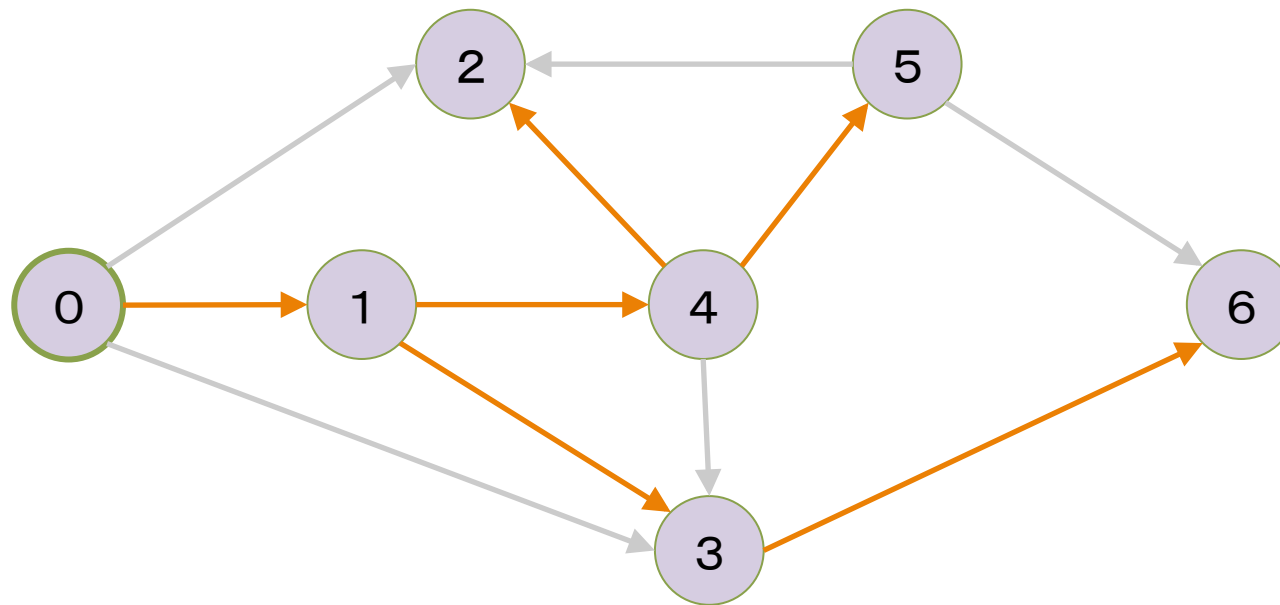
# トポロジカルソート 方法2

訪問順: 013631424541



# トポロジカルソート 方法2

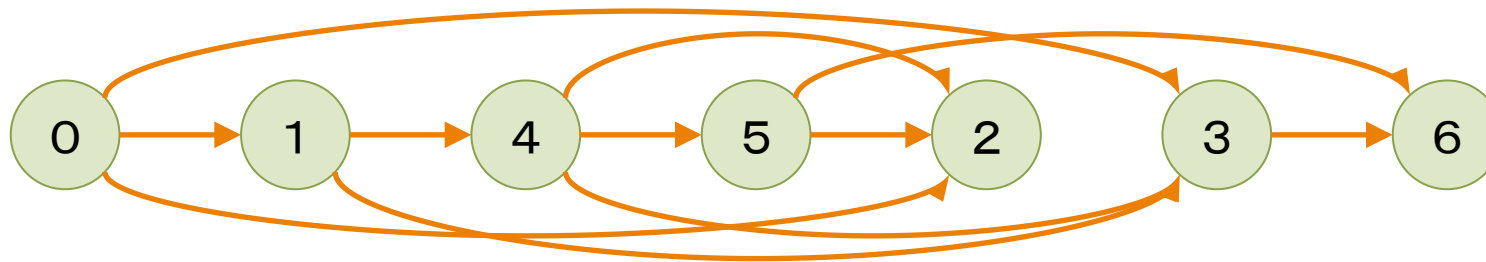
訪問順: 0 1 3 6 3 1 4 2 4 5 4 1 0



# トポロジカルソート 結果2

訪問順: 0136314245410

DFSで最後にその節点を訪問した順番を逆にする

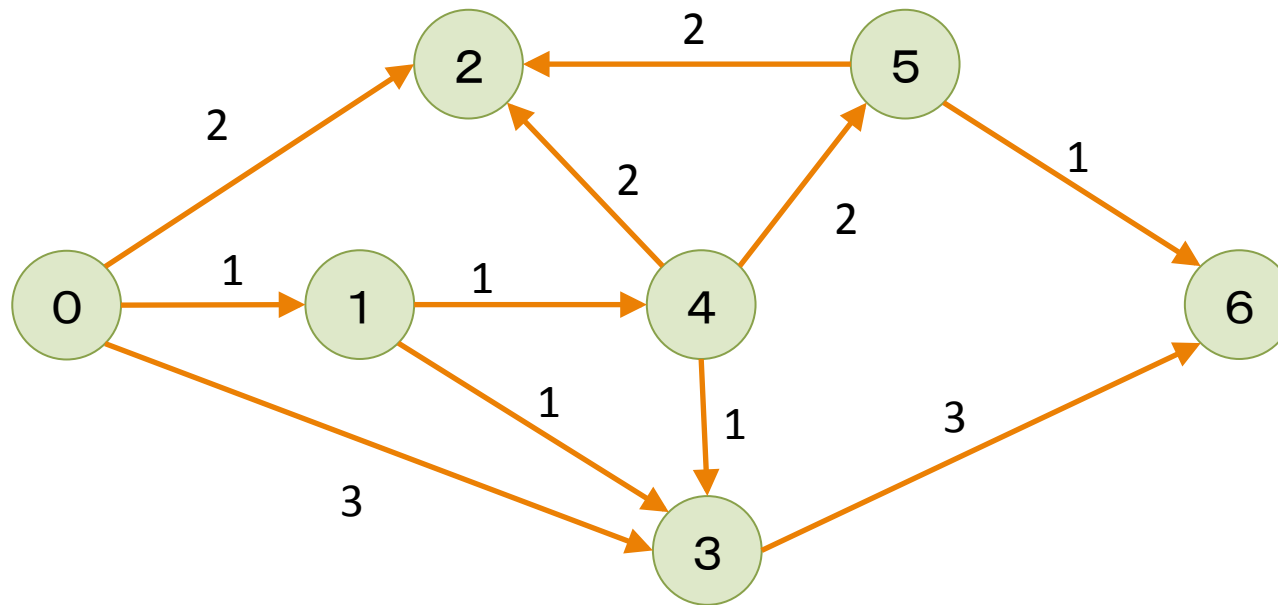




重みつきグラフの  
**最短経路**

# 重みつき最短経路

ダイクストラ法



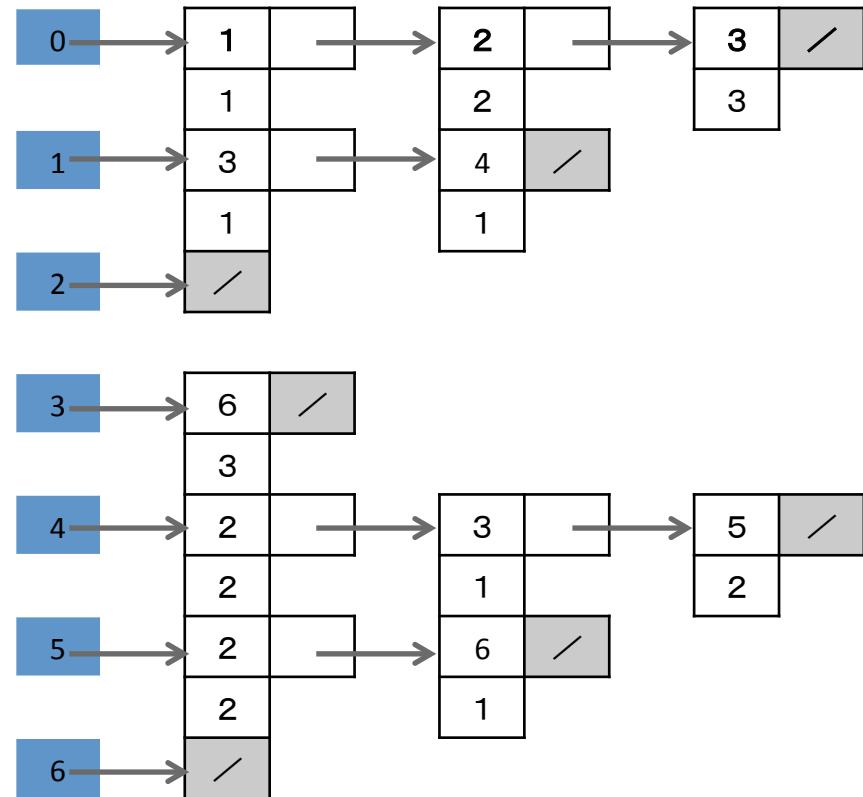
# ダイクストラ法 0

## 隣接行列

	d	接続1	d	接続2	d	接続3
0	0	-				
1	1	0				
2	2	0	4	4		
3	3	0	2	1	3	4
4	2	1				
5	4	4				
6	5	3	5	5		

d: 接続に示すノード経由での  
0からの距離  
接続ノードまでの距離と重みの和の最小値

## 重みつき隣接リスト



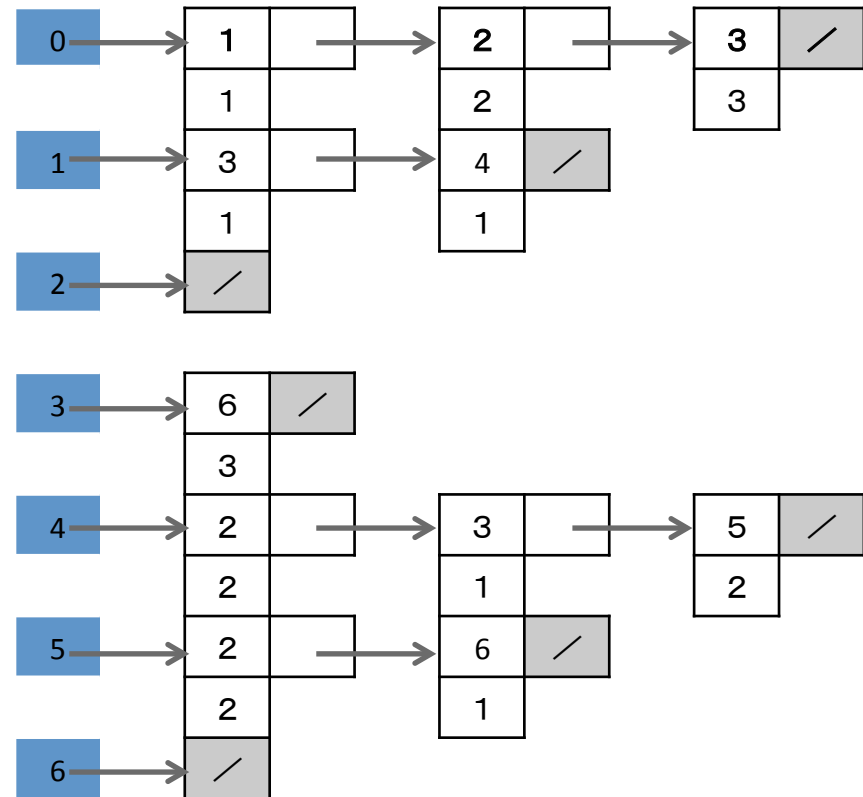
# ダイクストラ法 1

## 隣接行列

	d	接続
0	0	-
1	-1	-
2	-1	-
3	-1	-
4	-1	-
5	-1	-
6	-1	-

キュー
0

## 重みつき隣接リスト



# ダイクストラ法 2

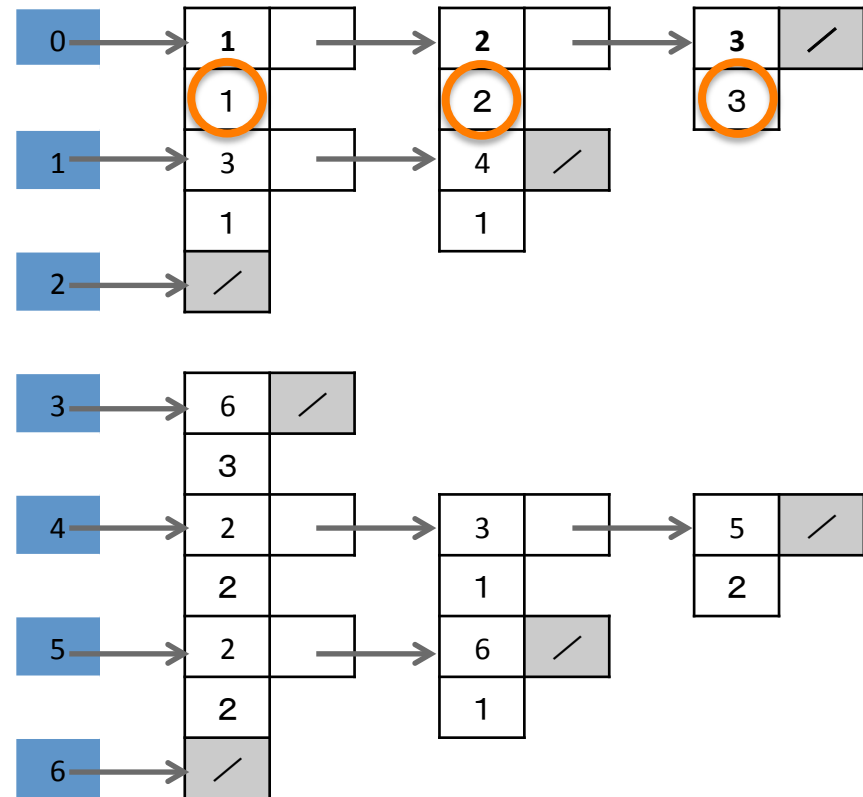
隣接行列

	d	接続
0	0	-
1	1	0
2	2	0
3	3	0
4	-1	-
5	-1	-
6	-1	-

キュー

0

重みつき隣接リスト



# ダイクストラ法 3

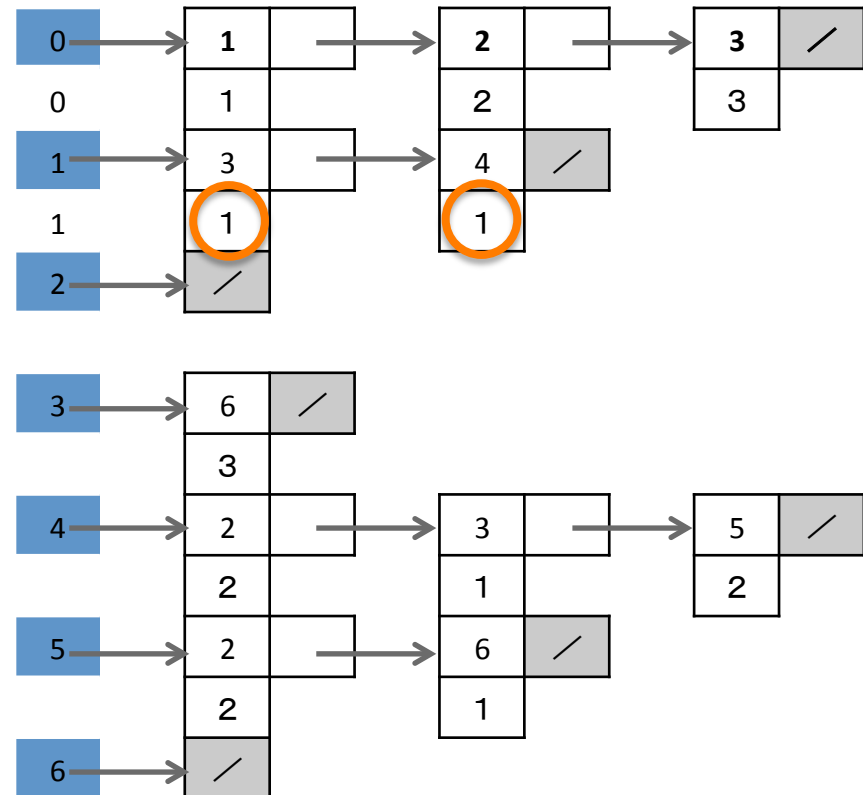
隣接行列

	d	接続
0	0	-
1	1	0
2	2	0
3	2	1
4	2	1
5	-1	-
6	-1	-

キュー

1
2
3

重みつき隣接リスト



# ダイクストラ法 4

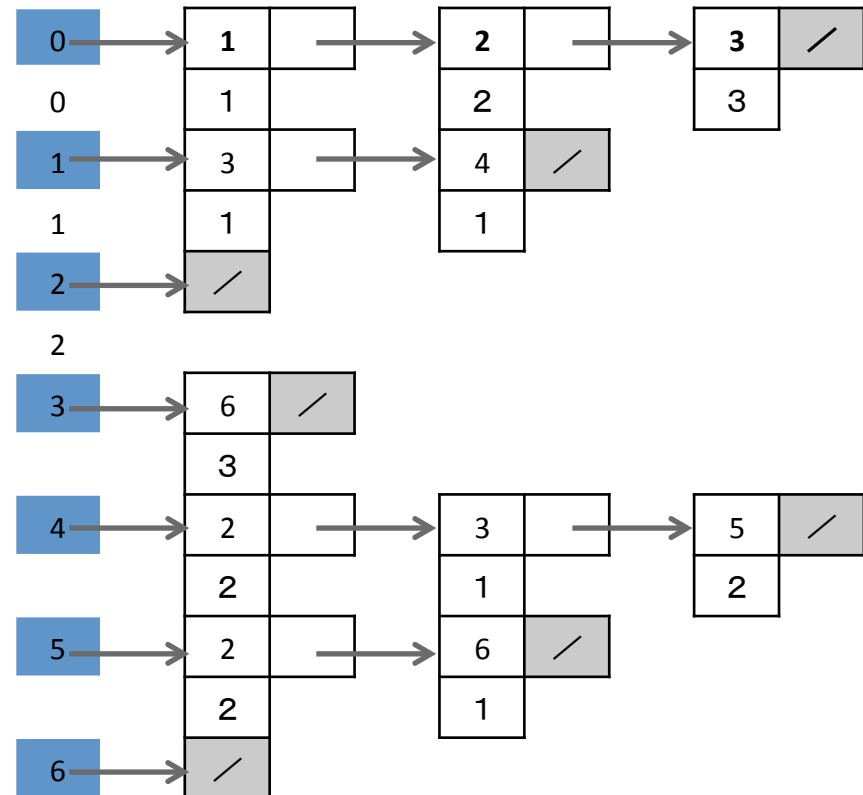
## 隣接行列

	d	接続
0	0	-
1	1	0
2	2	0
3	2	1
4	2	1
5	-1	-
6	-1	-

## キュー

キュー
2
3
4

## 重みつき隣接リスト



# ダイクストラ法 5

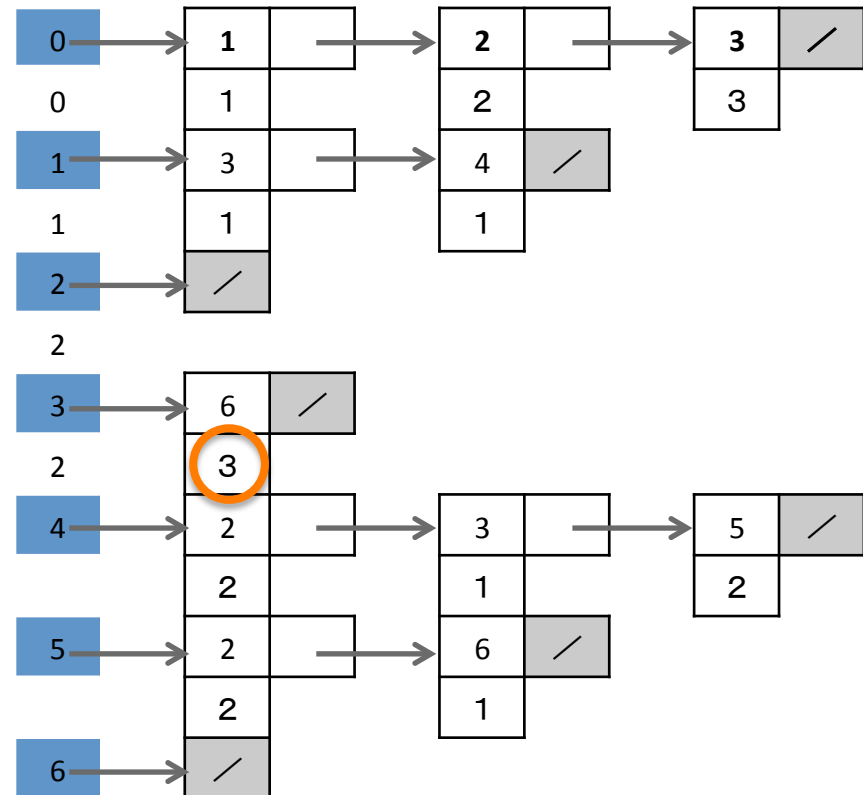
隣接行列

	d	接続
0	0	-
1	1	0
2	2	0
3	2	1
4	2	1
5	-1	-
6	5	3

キュー

3
4

重みつき隣接リスト





# ダイクストラ法 6

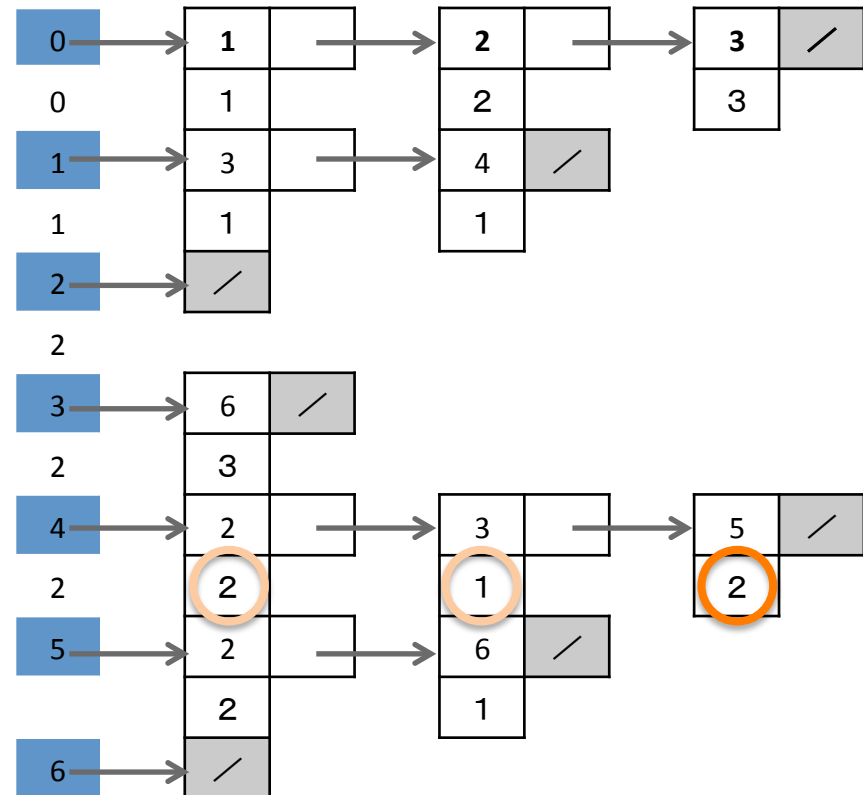
隣接行列

	d	接続
0	0	-
1	1	0
2	2	0
3	2	1
4	2	1
5	4	4
6	5	3

キュー

4
6

重みつき隣接リスト



# ダイクストラ法 7

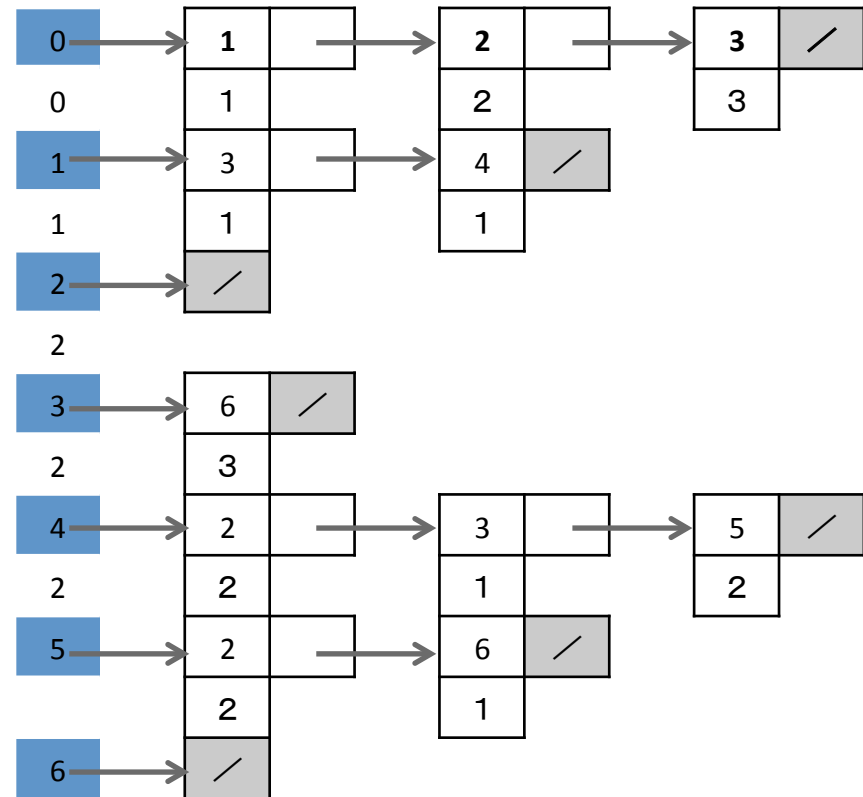
## 隣接行列

	d	接続
0	0	-
1	1	0
2	2	0
3	2	1
4	2	1
5	4	4
6	5	3

## キュー

6
5

## 重みつき隣接リスト



# ダイクストラ法 8

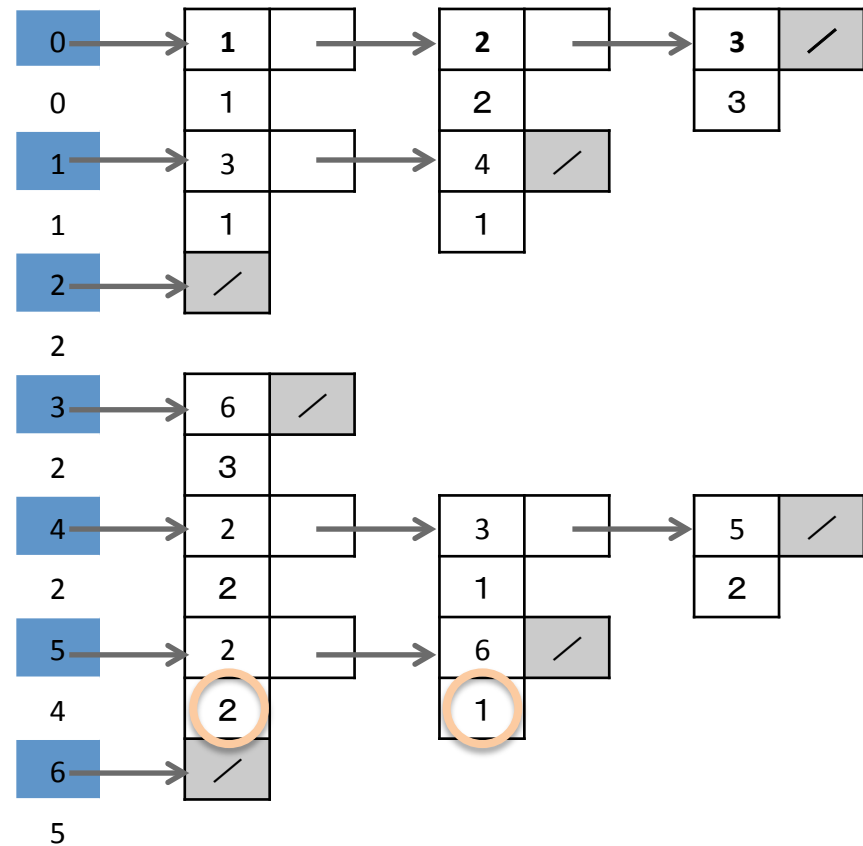
隣接行列

	d	接続
0	0	-
1	1	0
2	2	0
3	2	1
4	2	1
5	4	4
6	5	3

キュー

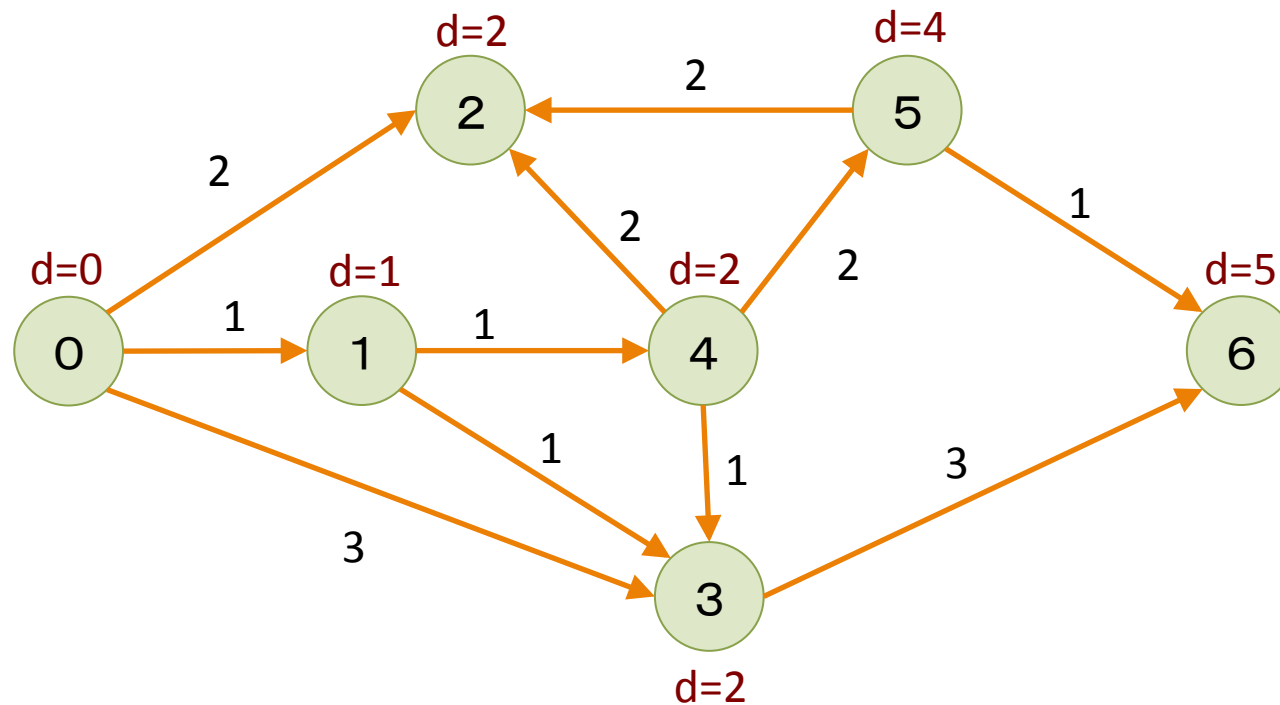
5

重みつき隣接リスト



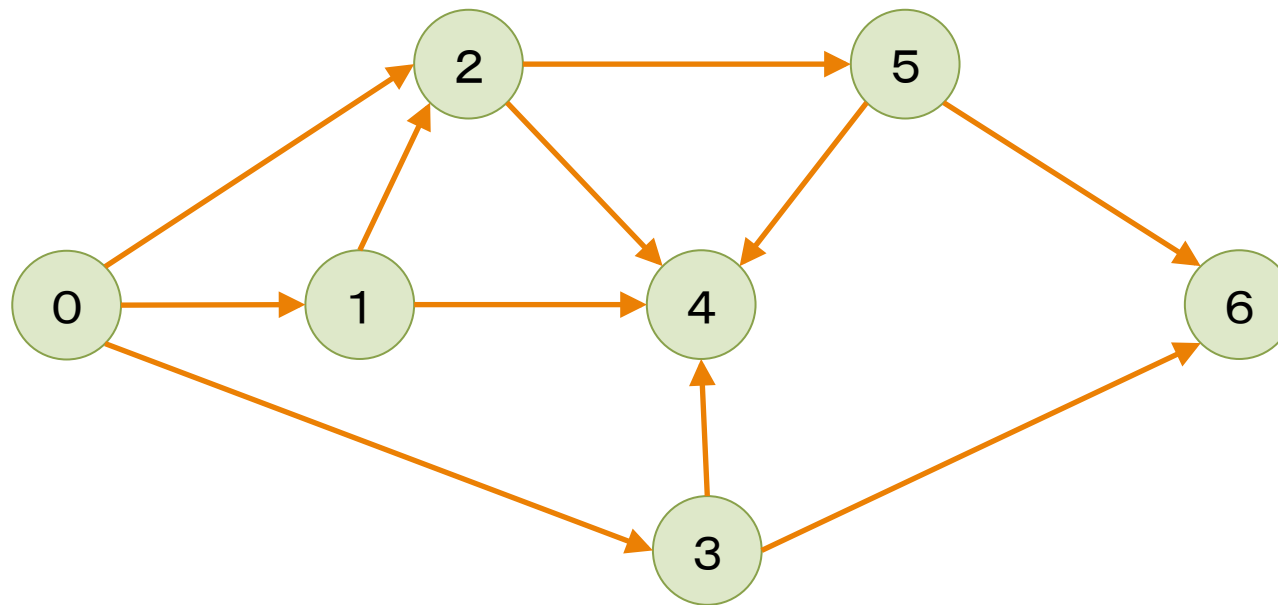
# 重みつき最短経路

節点0からの距離



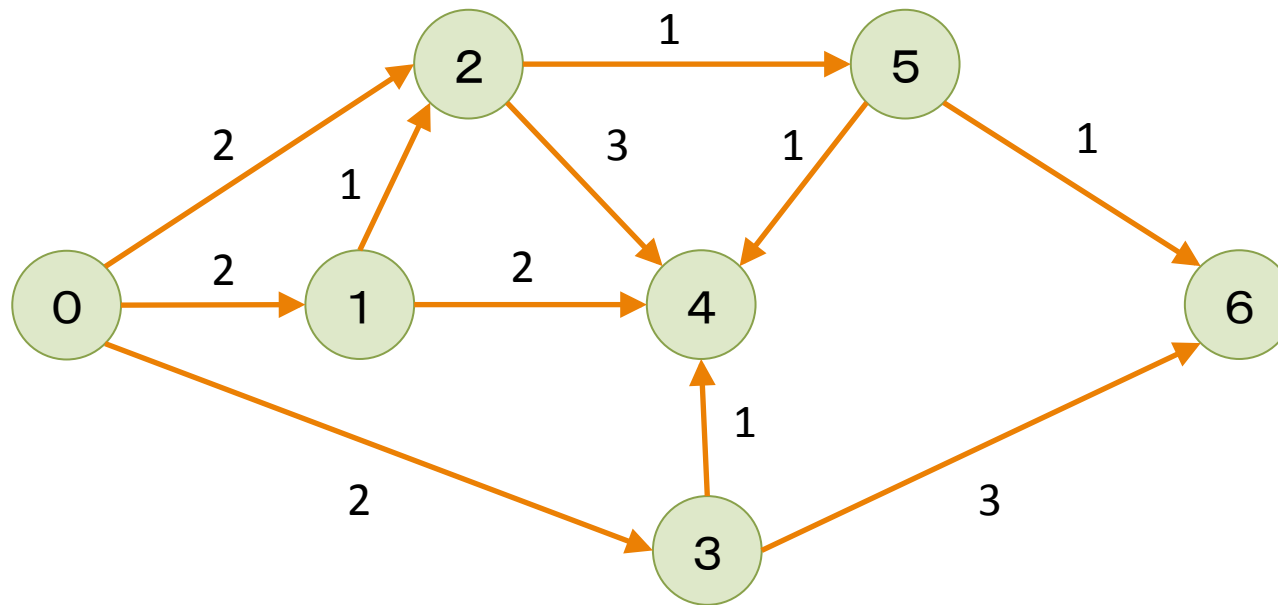
# やってみよう: 作業1

トポロジカルソート



# やってみよう: 作業2

節点0からの距離



# 連絡先

樋口文人

wenren@meiji.ac.jp