

第14回のキーワード

1

アルゴリズム関係

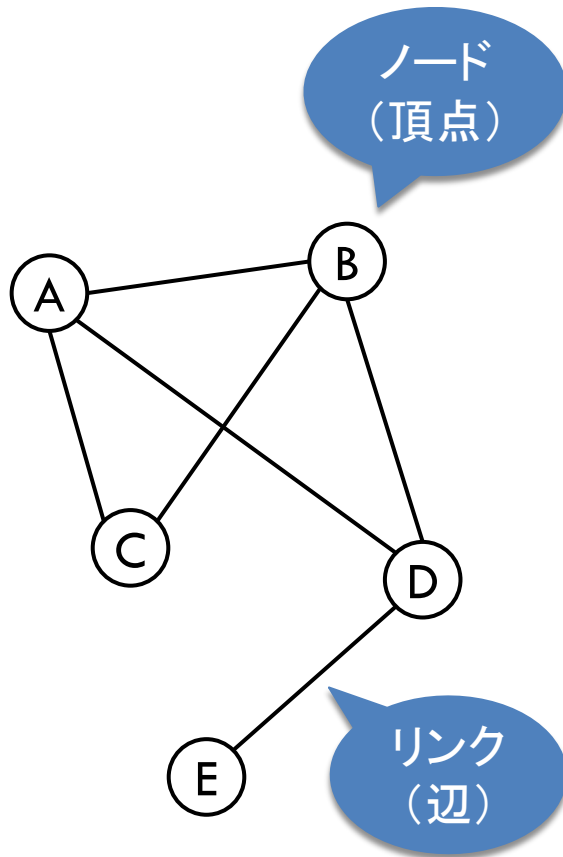
- グラフ構造
(graph structure)
- 無向グラフ
(undirected graph)
- 有向グラフ
(directed graph)
- 重み付きグラフ
(weighted graph)
- 隣接行列/リスト
(adjacency matrix/list)
- ダイクストラ法
(Dijkstra's algorithm)

Java関係

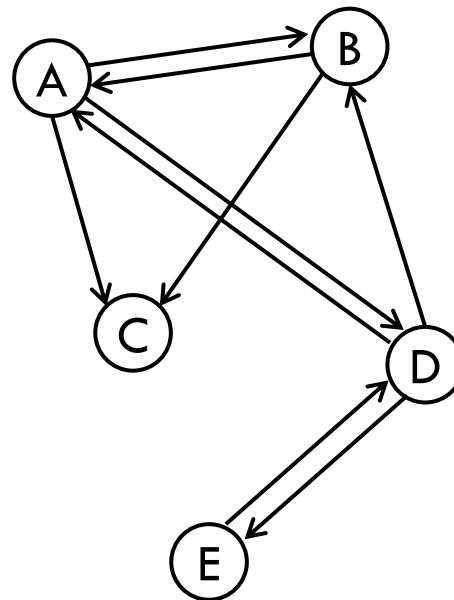
- コレクションフレームワーク
- List, ArrayList, LinkedList
- Set, HashSet, TreeSet
- Map, HashMap, TreeMap
- 匿名クラス
- 関数型インタフェース
- ラムダ式
- forEachメソッド

グラフ構造

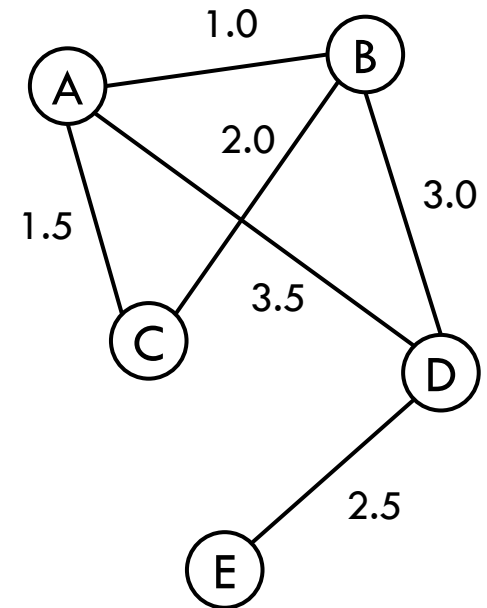
2



無向グラフ



有向グラフ



重み付きグラフ

隣接行列

3

□ ノード間の接続状態を要素とする行列

	A	B	C	D	E
A	0	1	1	1	0
B	1	0	1	1	0
C	1	1	0	0	0
D	1	1	0	0	1
E	0	0	0	1	0

無向グラフ

対称行列になる

$$a_{ij} = a_{ji}$$

	A	B	C	D	E
A	0	1	1	1	0
B	1	0	1	0	0
C	0	0	0	0	0
D	1	1	0	0	1
E	0	0	0	1	0

有向グラフ

一方通行があるので

$$a_{ij} = a_{ji} \text{とは限らない}$$

	A	B	C	D	E
A	∞	1.0	1.5	3.5	∞
B	1.0	∞	2.0	3.0	∞
C	1.5	2.0	∞	∞	∞
D	3.5	3.0	∞	∞	2.5
E	∞	∞	∞	2.5	∞

重み付きグラフ

リンクがない場合は
0または ∞ とする

最短経路問題

4

□ ダイクストラ法

- 重み付きグラフで、ある頂点から全頂点への最短経路を求める
- 理論的に、2点間の最短経路だけを求める確実な方法はない

□ アルゴリズム

- 基本方針：未確定の頂点から、1つずつ(始点からの)距離が最短のものを選んで確定済みにし、全頂点が確定したら終わり
 1. 最初は、全頂点の(始点からの)距離を未確定とする
 2. 全頂点が確定するまで以下を繰り返す
 - ① 未確定の頂点の中から、始点からの「仮の距離と経路」が最短のもの v を選択する(最初は始点を選択される)
 - ② v の距離と経路は確定済みとする
 - ③ v に直接連結している頂点の「仮の距離と経路」を更新する