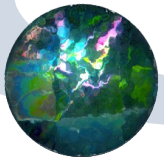
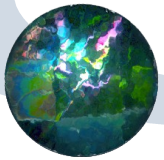


グラフを記述するための データ構造



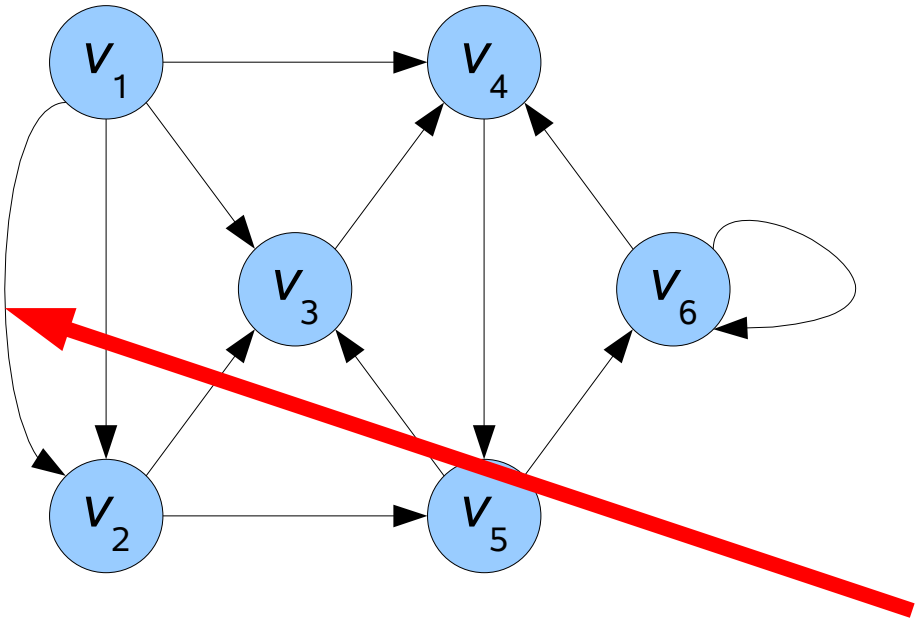
グラフを記述するには何が必要

- 探索問題を考えよう
  - 最短経路
  - 閉路探索
- 点から枝を伝って移動するための情報
  - 点を始点とする枝の一覧
- 枝の両端の点を知るための情報

隣接行列 (Adjacency Matrix)

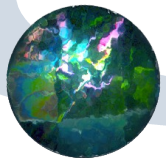
- 各要素：点間の連結の有無を示す

$$\partial^{+} a_i = v_j,\ \partial^{-} a_i = v_k \ \rightarrow\ \Gamma_{jk} = 1$$

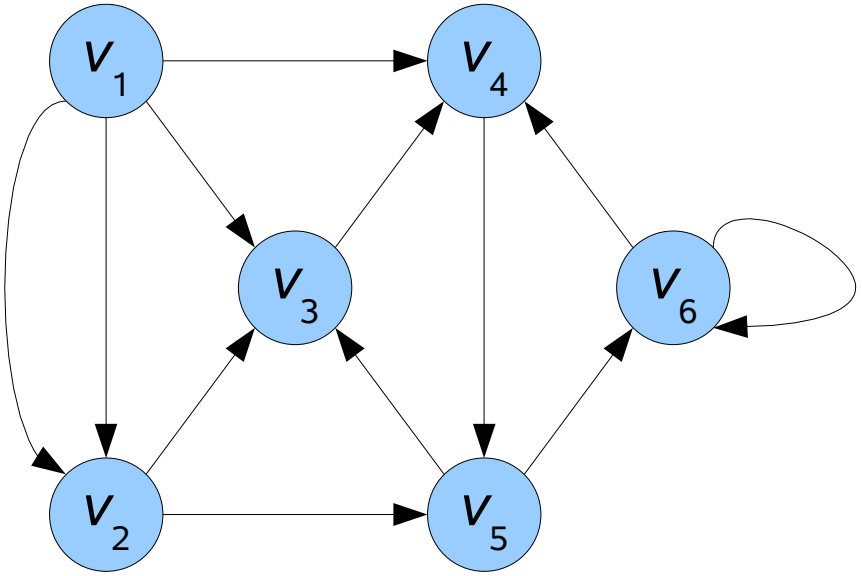


$$\Gamma = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

点v_1とv_2の間の並列弧の情報が表現されない。

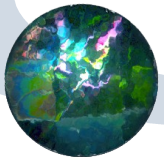


隣接行列の活用



$$\Gamma^2 = \begin{pmatrix} 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

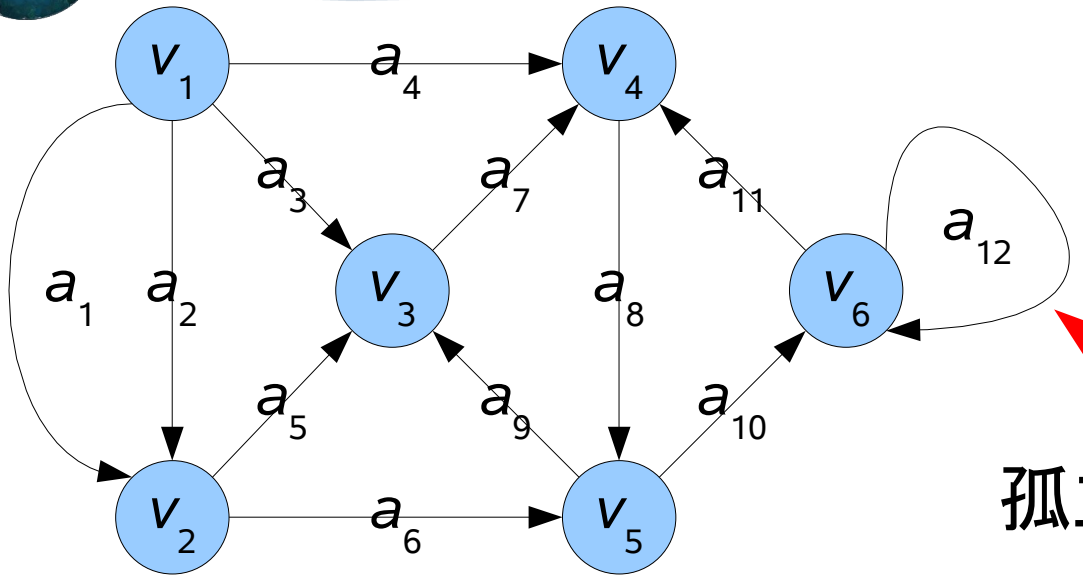
2回の移動で到達できる点



接続行列 (Incidence matrix)

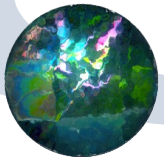
- 弧と点の関連付け

$$D(v,a)=\begin{cases} +1 & \text{if} \quad \partial^{+}a=v \wedge \partial^{-}a \neq v \\ -1 & \text{if} \quad \partial^{-}a=v \wedge \partial^{+}a \neq v \\ 0 & \text{otherwise} \end{cases}$$



孤立弧a_{12}の情報が無い

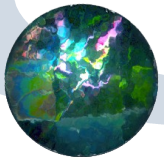
$$D=\begin{pmatrix} +1 & +1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & +1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & +1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & +1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 & 0 \end{pmatrix}$$



接続行列の活用：グラフ中の定常的流れ

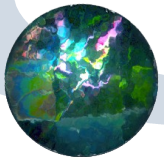
- 枝集合上の関数　　$\varphi : A \to R$

- 定常的な流れ　　$\left(D\varphi\right)_v = \sum_a D(v,a)\varphi(a) = 0$
  - 各点へ流れ込む量と流れ出す量がつり合う
- 接続行列はこうやって使えるが?



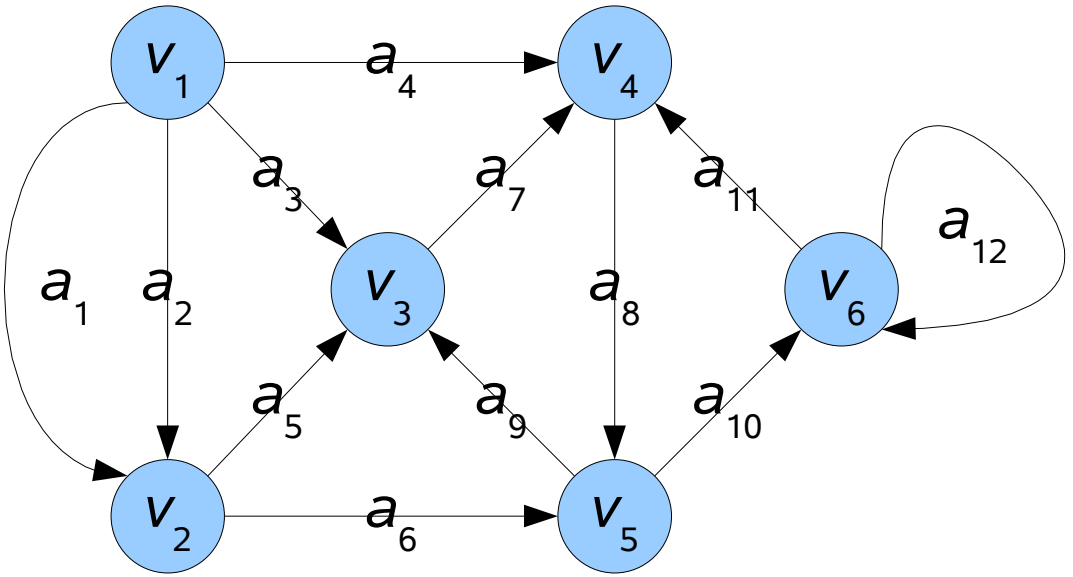
行列表現の問題点

- 演算という観点では便利
- ほとんどの要素がゼロ
  - 情報量が少ない：sparse (薄い、貧弱な)
- 隣接行列は並列弧を表現出来ない
- 接続行列は孤立弧を表現出来ない

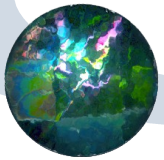


リスト表現の利点

- 現代的なプログラミング言語はもっと多様なデータ構造が扱えるではないか
  - 柔軟なデータ構造とアルゴリズム
- 各頂点を始点とする弧の一覧（リスト）を作成する
  - 必要な分だけの情報
  - 並列弧・孤立弧が両方とも扱える
- 弧は始点と終点の情報を知っている



$$\delta^{+} v_1 = \{a_1, a_2, a_3, a_4\}$$
$$\delta^{+} v_2 = \{a_5, a_6\}$$
$$\delta^{+} v_3 = \{a_7\}$$
$$\delta^{+} v_4 = \{a_8\}$$
$$\delta^{+} v_5 = \{a_9, a_{10}\}$$
$$\delta^{+} v_6 = \{a_{11}, a_{12}\}$$



再帰的探索が可能となる

ある頂点vに対する関数 $f(v)$ {

forall (頂点v を始点とする辺a){

頂点 w はa の終点

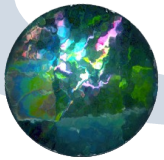
$f(w)$

}

}

注意

- 終了の条件
- 同じところをグルグル回らないためのチェック



Javaでの実装

- 点のクラス
  - 点の名前
  - 点を始点とする弧の一覧
- 弧のクラス
  - 弧の名前
  - 弧の始点と終点