# リンク解析
# (Link Analysis)

**PageRank**

# リンク解析
(Link Analysis)

**PageRank**

The $25,000,000,000 eigenvector

DOI:10.1145/3434642

**Ensuring the success of big graph processing for the next decade and beyond.**

BY SHERIF SAKR, ANGELA BONIFATI, HANNES VOIGT, AND ALEXANDRU IOSUP

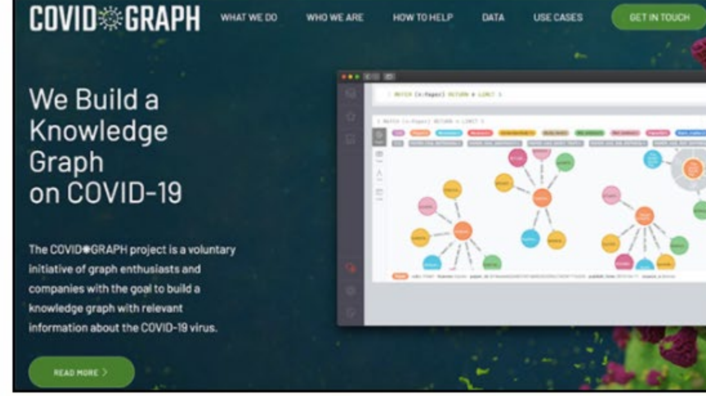
# The Future Is Big Graphs: A Community View on Graph Processing Systems

GRAPHS ARE, BY nature, 'unifying abstractions' that can leverage interconnectedness to represent, explore, predict, and explain real- and digital-world phenomena. Although real users and consumers of graph instances and graph workloads understand these abstractions, future problems will require new abstractions and systems. What needs to happen in the next decade for big graph processing to continue to succeed?

Comm. ACM 2021


**Contact Tracing**

In the Geneva canton, Switzerland, the Office of the Surgeon General collects all results from laboratories performing SARS-CoV-2 testing.




**Covid Graph**

DZD and Researchers Connecting COVID-19 Publications to Drug Repurposing


**Project Domino**

COVID-19 Intervention Project Tracking Misinformation to Educate Local Communities

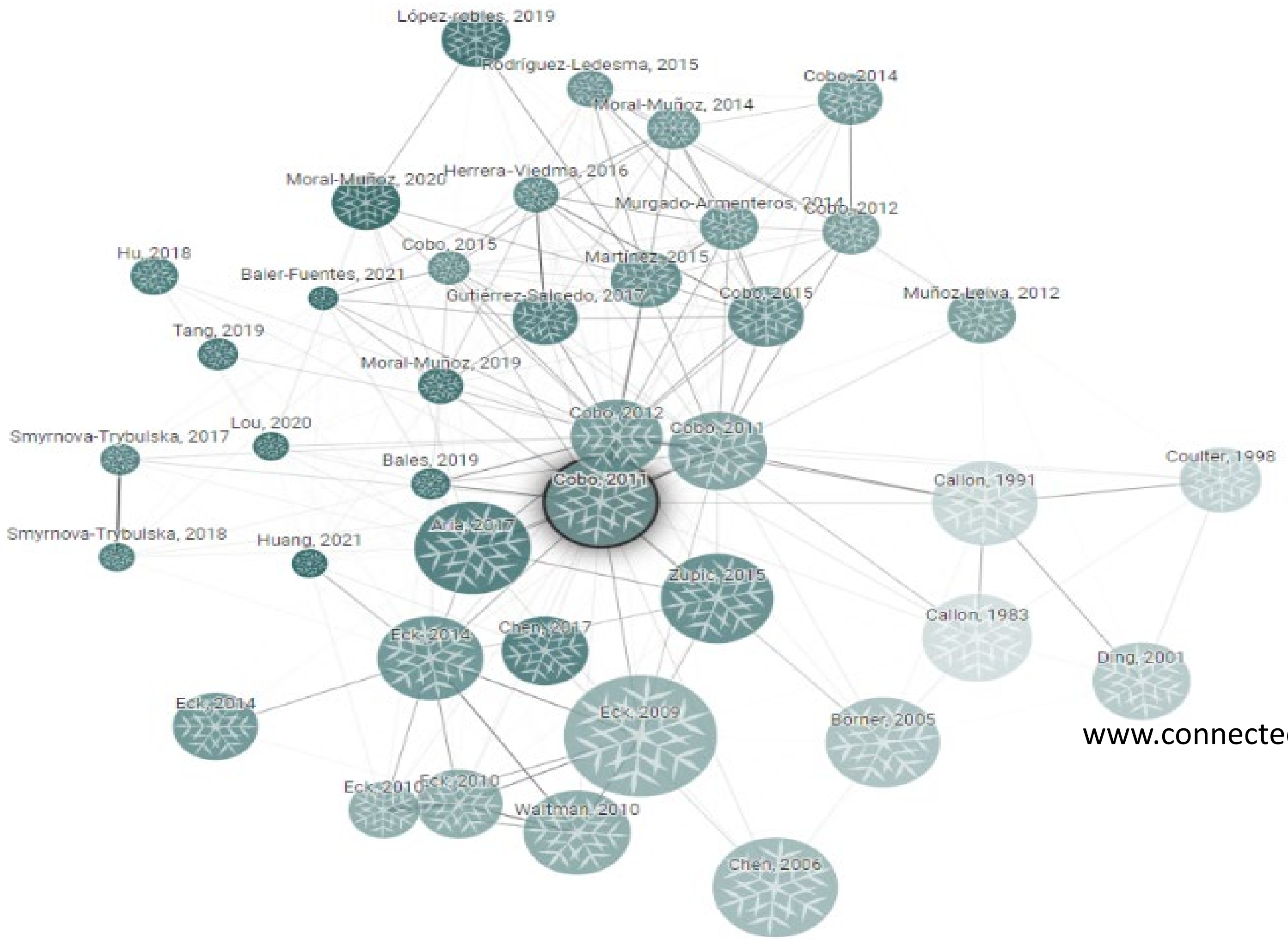

**Hume-Covid**

Using GraphAware Hume for COVID-19 Contact Tracing and Smart Quarantine

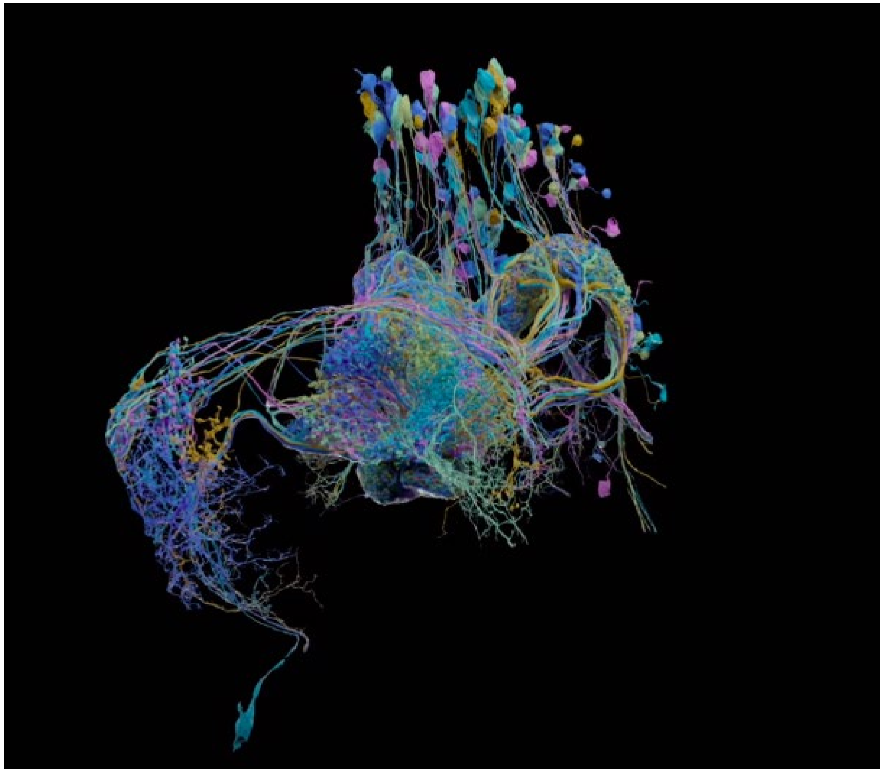
## Get Involved in Graphs 4 COVID-19

https://neo4j.com/graphs4good/covid-19/

# グラフデータ分析：コミュニティ抽出・可視化



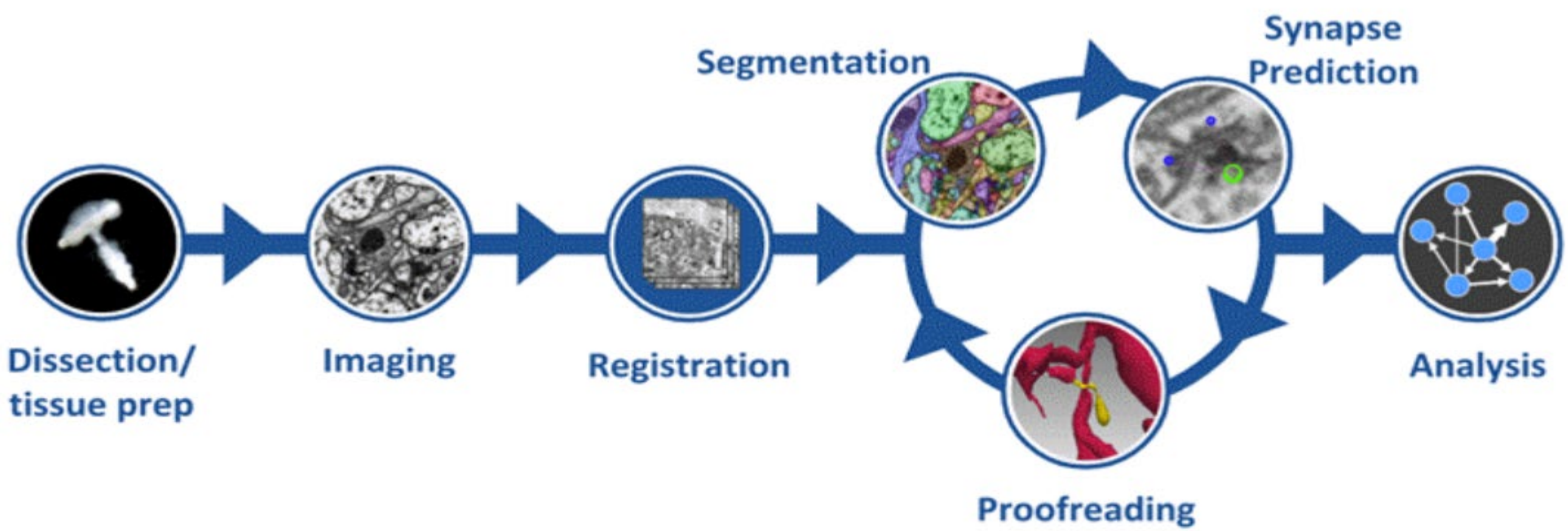

www.connectedpapers.com

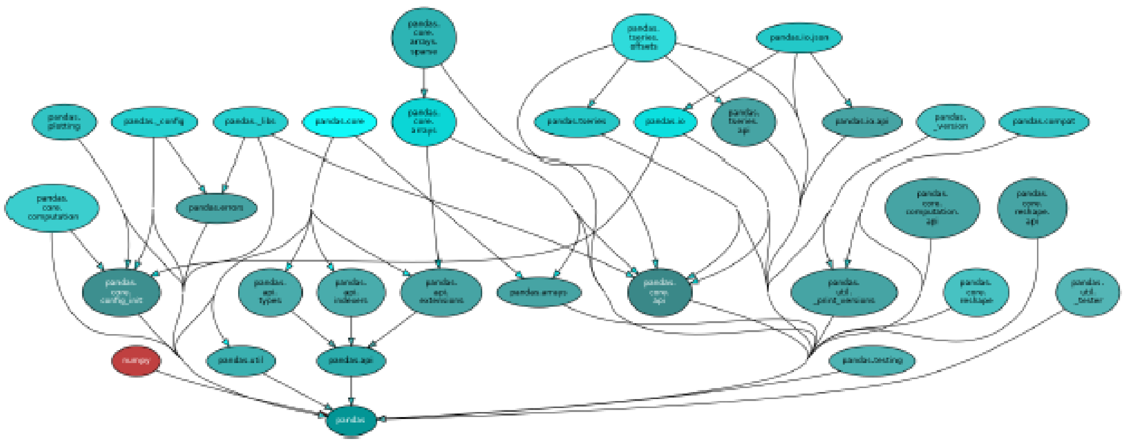


Neuron network of a fly’s brain

# コネクトーム（神経系の接続マップ）の分析・可視化

https://www.janelia.org/project-team/flyem/hemibrain






Python code module dependency network

# リンク解析
(Link Analysis)

**PageRank**

The $25,000,000,000 eigenvector

# 用語スパム（Term Spam）の問題

- 用語スパム(term spam)の問題

# 用語スパム（Term Spam）の問題

- 用語スパム(term spam)の問題

リンクスパマーは，検索をほとんど無用なものにしていた

# 用語スパム（Term Spam）の問題

- 用語スパム(term spam)の問題
  リンクスパマーは，検索をほとんど無用なものにしていた

- 用語スパムと戦う PageRank
  ランダムサーファーモデル(Random Surfer Model)：

＊ウェブ上で多くのランダムサーファーの挙動のシミュレーション

＊各ランダムサーファーはランダムなページからスタートし，現在のページがリンクしているページの1つにランダムに跳ぶ

＊このプロセスを数多く繰り返すことができれば，サーファー達はあるページ群に集まる傾向にある

＊多数のサーファーが滞在しているページ（PageRank大）は滅多に訪問されないページ（PageRank小）よりも重要であると考えられる

# 用語スパム（Term Spam）の問題

- 用語スパム(term spam)の問題
  リンクスパマーは，検索をほとんど無用なものにしていた
- 用語スパムと戦う PageRank
  ランダムサーファーモデル(Random Surfer Model)：
  ＊ウェブ上で多くのランダムサーファーの挙動のシミュレーション
  ＊各ランダムサーファーはランダムなページからスタートし，現在のページがリンクしているページのうちにランダムに跳ぶ
  ＊このプロセスを数多く繰り返すことができれば，サーファー達はあるページ群に集まる傾向にある
  ＊多数のサーファーが滞在しているページ（PageRank大）は滅多に訪問されないページ（PageRank小）よりも重要であると考えられる

サーファーがあるページにいる確率の極限値がそのページのPageRankです

人々は有用だと考えるページにリンクを張る傾向があり，そのため
ランダムサーファーは有用なページにいる傾向にある

サーチクエリーに対して，最初に重要なページを出す

# 用語スパム（Term Spam）の問題

- 用語スパムと戦う

＊ページの内容は，そのページに出現する用語だけではなく，
そのページを指しているリンクの中で使われている用語で
判断されている
＊シャツ販売者自身が自分のことを言っていることよりも他の
ページがその販売者について言っていることをGoogleが信じる

ランダムサーファーのシミュレーションで
ページの重要性が近似できる理由

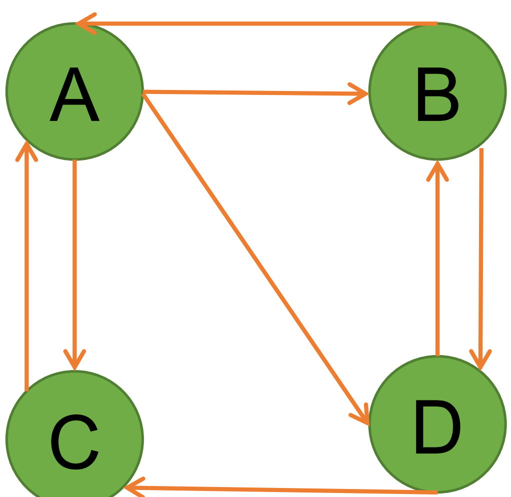
- ランダムサーファーの挙動はウェブユーザーが訪問しそうな
ページを示している
- ユーザーは無用なページよりも有用なページをより訪問しやすい

# PageRankの定義

- PageRank
  ウェブの各ページに1つの実数値を割り当てる関数
  （高ければ高いほどそのページはより重要である）

# PageRankの定義

- PageRank
  ウェブの各ページに1つの実数値を割り当てる関数
  （高ければ高いほどそのページはより重要である）
- ウェブの世界を有向グラフとして考える




遷移行列(transition matrix)

$$
\begin{array}{c}
 \\ A \\ B \\ C \\ D
\end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\begin{bmatrix}
0 & 1/2 & 1 & 0 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 1/2 & 0 & 0
\end{bmatrix}
\end{array}
= \mathbf{M}
$$

# PageRankの定義

- ランダムサーファーの位置の確率分布は，サーファーがページ j にいる確率が j 番目の要素であるような列ベクトル **v** によって記述できる

  ＊　この確率はPageRank関数である

- ランダムサーファーがウェブの k 個のページの任意のページから同じ確率でスタートする

$$\mathbf{v}_0 = \begin{bmatrix} 1/k \\ 1/k \\ \cdots \\ 1/k \end{bmatrix}$$

# PageRankの定義

- 1ステップ後のサーファーの分布

$$\mathbf{v}_i = \mathbf{M}\mathbf{v}_{i-1}$$

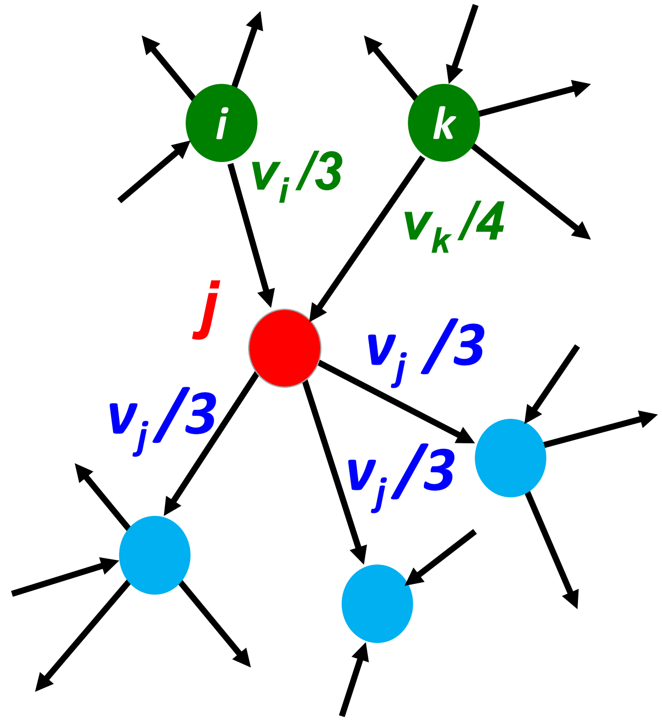
$$\mathbf{v}_0 = \begin{bmatrix} 1/k \\ 1/k \\ \cdots \\ 1/k \end{bmatrix} \qquad \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} = \mathbf{M}$$

遷移行列
(transition matrix)

# PageRankの定義

- 1ステップ後のサーファーの分布

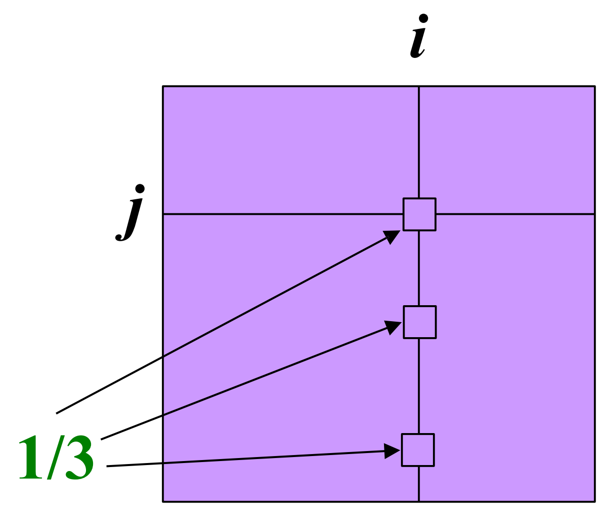
$v_j = v_i / 3 + v_k / 4$




$$v_j = \sum_{i \to j} \frac{v_i}{d_i}$$

$d_i$ : out-degree of node $i$

出次数

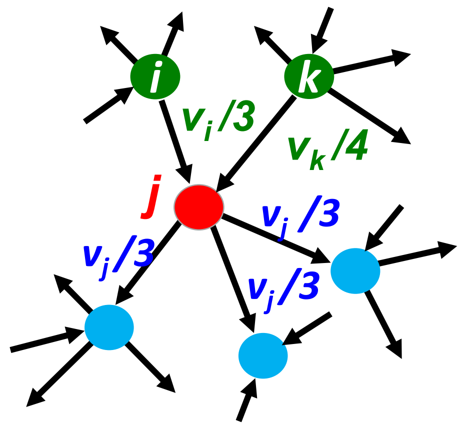



**M**

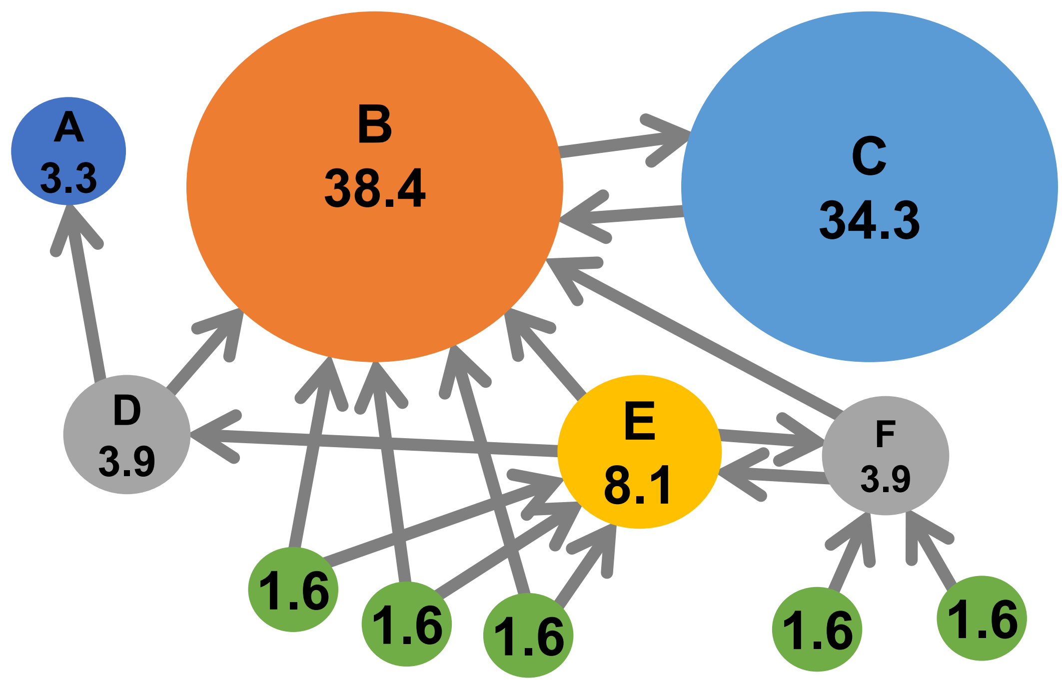
# PageRankの定義

- 1ステップ後のサーファーの分布

$v_j = v_i / 3 + v_k / 4$




- A “vote” from an **important** page is worth more
- A page is **important** if it is pointed to by other important pages



# PageRankの定義

- 1ステップ後のサーファーの分布

$$\boxed{\mathbf{v}_i = \mathbf{M}\mathbf{v}_{i-1}}$$

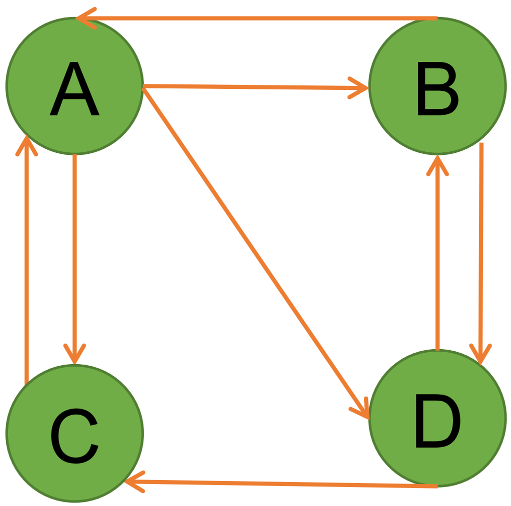
- i ステップ後のサーファーの分布

$$\mathbf{v}_i = \mathbf{M}^i \mathbf{v}_0$$

$$\mathbf{v}_0 = \begin{bmatrix} 1/k \\ 1/k \\ \dots \\ 1/k \end{bmatrix} \qquad \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} = \mathbf{M}$$

遷移行列
(transition matrix)

# PageRank計算の例



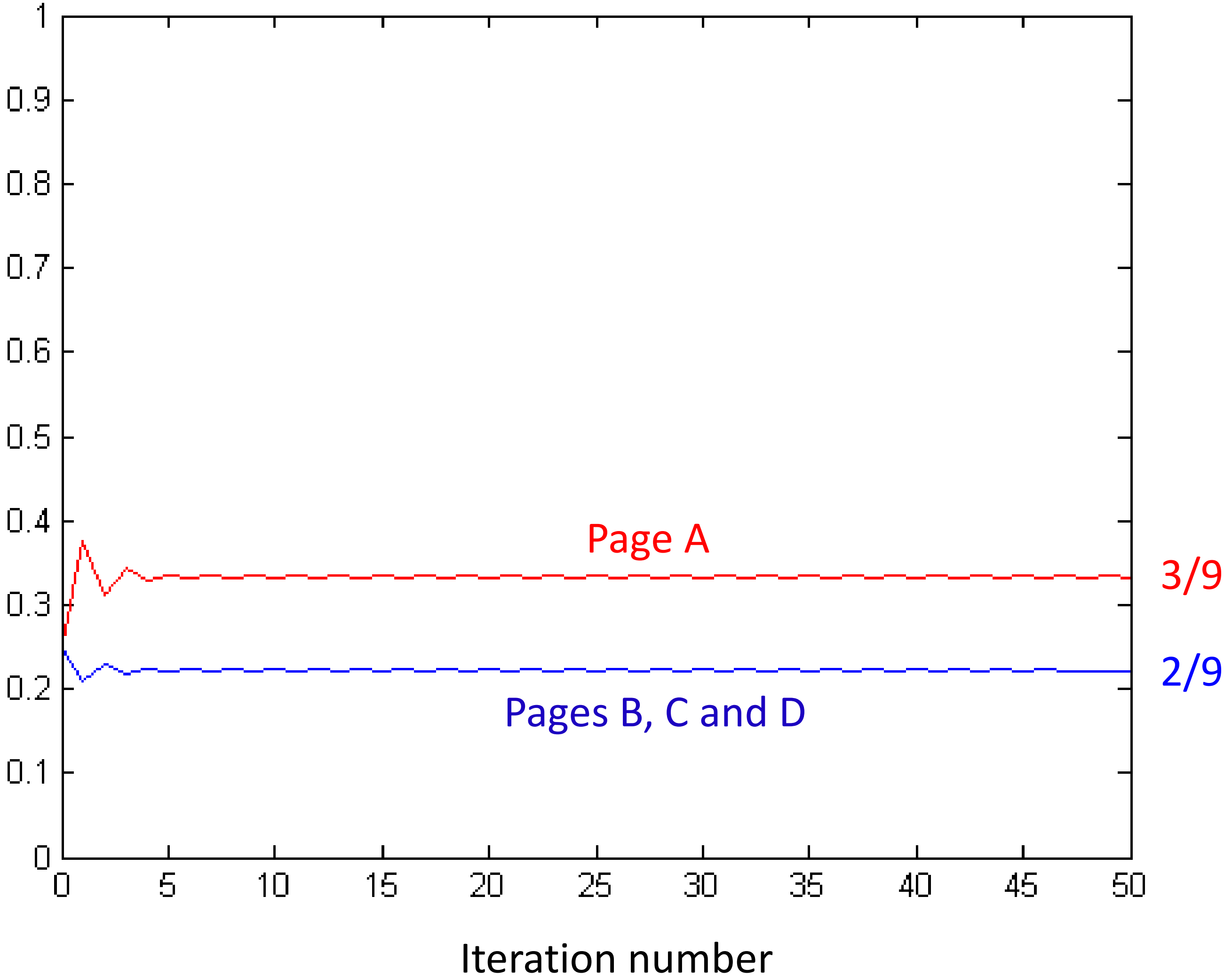

$$\mathbf{v}_0 = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}^T$$

$$\mathbf{v}_1 = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

$$\mathbf{v}_2 = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix} = \begin{bmatrix} 15/48 \\ 11/48 \\ 11/48 \\ 11/48 \end{bmatrix}$$

# PageRank計算の例



# 固有ベクトル計算としてのPageRank

- マルコフ過程(Markov process)

グラフは強連結(strongly connected) である場合，サーファーの分布は，$\boldsymbol{v} = \boldsymbol{M}\boldsymbol{v}$ を満たす分布 $\boldsymbol{v}$ の極限に近づく

強連結: 任意のノードから他の任意のノードに到達することが可能である

# 固有ベクトル計算としてのPageRank

- マルコフ過程(Markov process)

グラフは強連結(strongly connected) である場合，サーファーの分布は，***v* = *Mv*** を満たす分布 ***v*** の極限に近づく

強連結：任意のノードから他の任意のノードに到達することが可能である

- 分布は ***M*** をもう一度かけても変化しないとき，極限に到達している　➞　極限の ***v*** は M の固有ベクトルである

$$\boxed{\boldsymbol{Mv} = \lambda \boldsymbol{v}} \qquad \lambda = 1$$

- M は確率行列(stochastic matrix)なので，v は主固有ベクトル(principal eigenvector)である

確率行列：各列は足し合わせると1となる

- M は確率行列　→　主固有ベクトルに関連する固有値は1である

# 固有ベクトル計算としてのPageRank

- M の主固有ベクトル v は，長い時間が経った後，サーファーのいる確率が最も高い場所を教えてくれる

- PageRank　サーファーがあるページに存在する可能性が高ければ高いほど，そのページはより重要である

- M の主固有ベクトルを，次のように計算できる

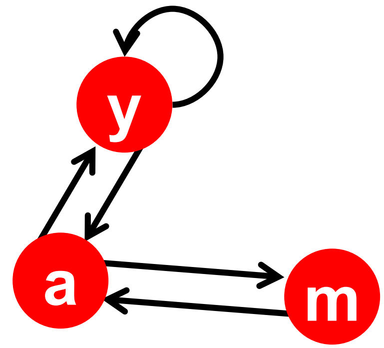
## べき乗法(Power iteration method)

- Suppose there are *N* web pages
- Initialize: $\mathbf{v}^{(0)} = [1/N, \ldots., 1/N]^T$
- Iterate: $\mathbf{v}^{(t+1)} = \mathbf{M} \cdot \mathbf{v}^{(t)}$
- Stop when $|\mathbf{v}^{(t+1)} - \mathbf{v}^{(t)}| < \varepsilon$

# 固有ベクトル計算としてのPageRank

練習

- **Power Iteration:**
  - Set $v_j = 1/N$
  - **1:** $v'_j = \sum_{i \to j} \frac{v_i}{d_i}$
  - **2:** $\boldsymbol{v} = \boldsymbol{v}'$
  - Goto **1**



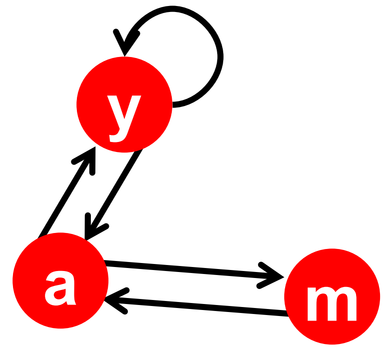

| | y | a | m |
|---|---|---|---|
| y | | | |
| a | | | |
| m | | | |

**M**

Iteration 0, 1, 2, …

# 固有ベクトル計算としてのPageRank

- **Power Iteration:**
  - Set $v_j = 1/N$
  - **1:** $v'_j = \sum_{i \to j} \frac{v_i}{d_i}$
  - **2:** $\boldsymbol{v} = \boldsymbol{v}'$
  - Goto **1**




| | y | a | m |
|---|---|---|---|
| y | ½ | ½ | 0 |
| a | ½ | 0 | 1 |
| m | 0 | ½ | 0 |

**M**

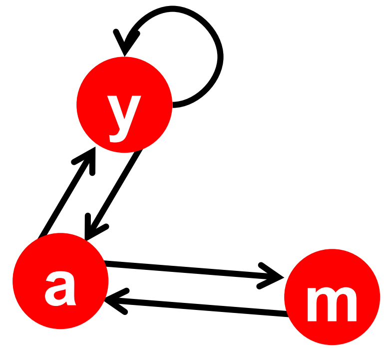
- **Example:**

$$\begin{bmatrix} v_y \\ v_a \\ v_m \end{bmatrix} = \begin{array}{cccccc} 1/3 & 1/3 & 5/12 & 9/24 & & 2/5 \\ 1/3 & 3/6 & 1/3 & 11/24 \ldots & & 2/5 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 1/5 \end{array}$$

Iteration 0, 1, 2, …

# 固有ベクトル計算としてのPageRank

- **Power Iteration:**
  - Set $v_j = 1/N$
  - **1:** $v'_j = \sum_{i \to j} \frac{v_i}{d_i}$
  - **2:** $\boldsymbol{v} = \boldsymbol{v}'$
  - Goto **1**




| | y | a | m |
|---|---|---|---|
| y | ½ | ½ | 0 |
| a | ½ | 0 | 1 |
| m | 0 | ½ | 0 |

**M**

- **Example:**

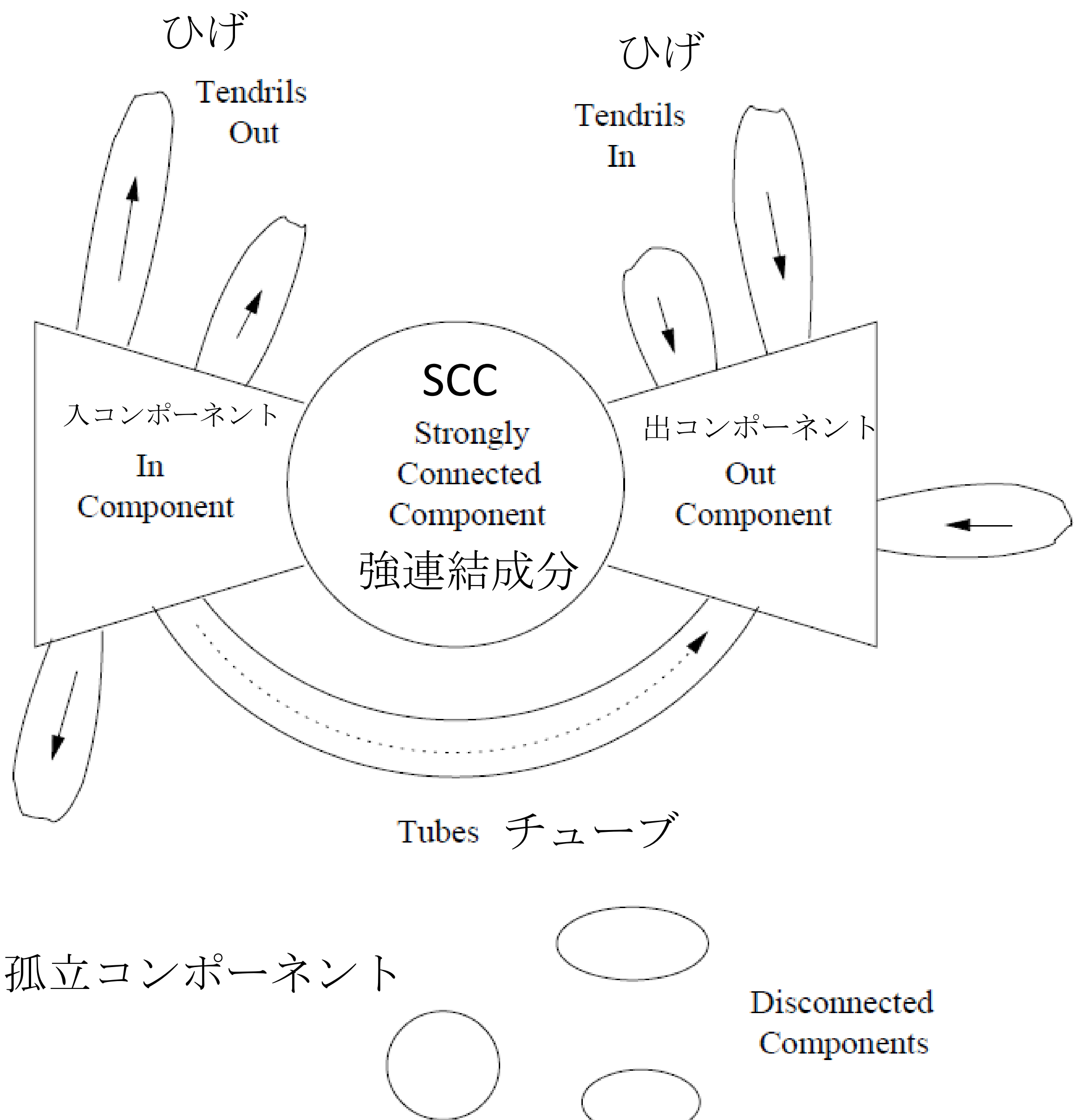
$$\begin{bmatrix} v_y \\ v_a \\ v_m \end{bmatrix} = \begin{array}{cccccc} 1/3 & 1/3 & 5/12 & 9/24 & & 2/5 \\ 1/3 & 3/6 & 1/3 & 11/24 & \ldots & 2/5 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 1/5 \end{array}$$

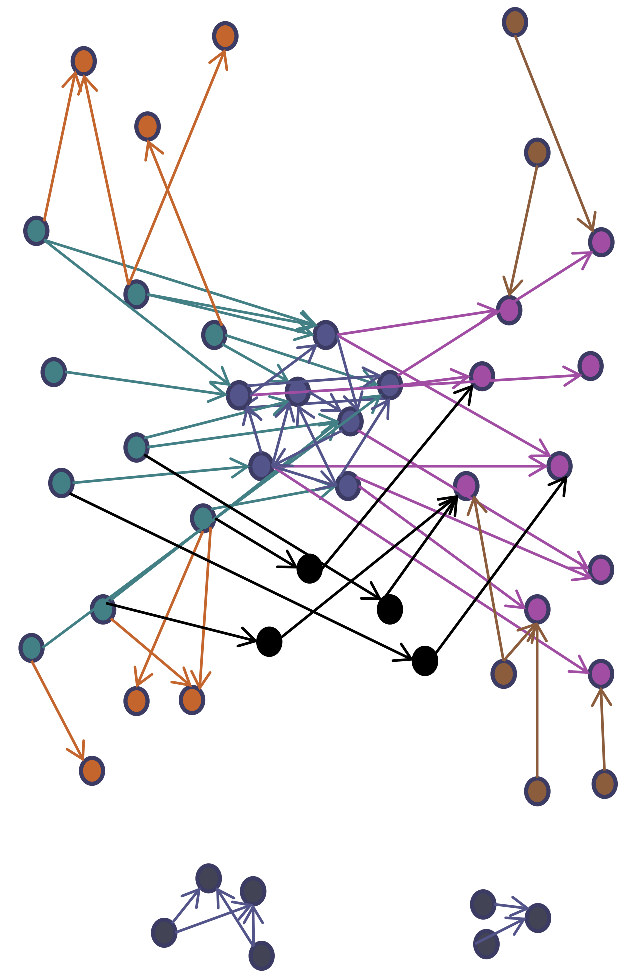
**練習:**

**v = Mv** を
ガウスの消去法
で解く

$$v_y = v_y/2 + v_a/2$$
$$v_a = v_y/2 + v_m$$
$$v_m = v_a/2$$
$$v_y + v_a + v_m = 1$$

# ウェブの”蝶ネクタイ”(bowtie)の構造



# 固有ベクトル計算としてのPageRank

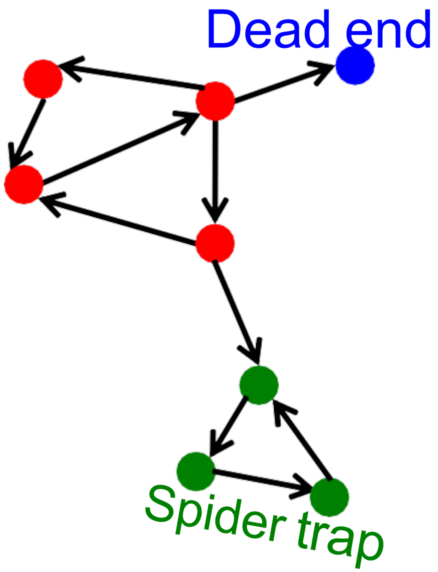
- そのような不都合を避けるためにPageRankは修正される

- 避けなければならない問題は2つある

（1） 行き止まり（Dead Ends）

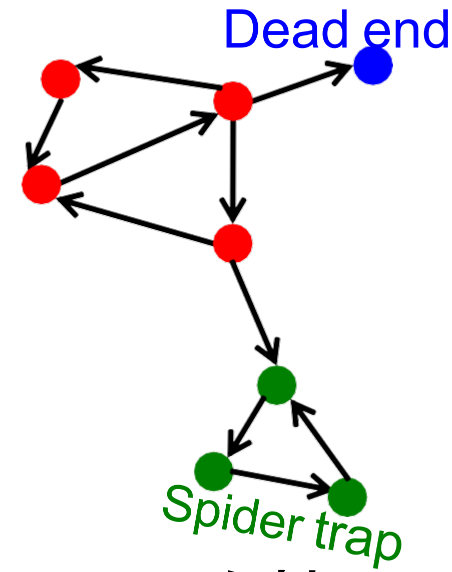
＊ リンクが出ていないページ

＊ サーファーが消えてしまう



# 固有ベクトル計算としてのPageRank

- そのような不都合を避けるためにPageRankは修正される




- 避けなければならない問題は2つある

（1） 行き止まり（Dead Ends）

　＊ リンクが出ていないページ

　＊ サーファーが消えてしまう

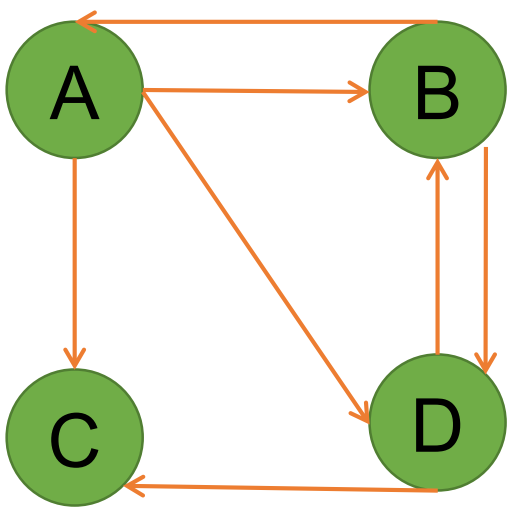
（2） スパイダートラップ(Spider Trap)

　＊ そのページ群のすべてのページは出リンクを持っているが，そのページ群以外のどの他のページにはリンクしていない

- これらの両方の問題ともテレポート(teleport)と呼ばれる手法で解決することができる

# 行き止まり（Dead Ends）

- 行き止まりを許せば，ウェブの遷移行列は確率的ではない（列のいくつかは，足し合わせると1ではなくなる）
- ベキ乗を増やしながら$M^i v$計算すれば結果は0になる
- ウェブの重要性は流れ去り，ページの相互の重要性に関する情報を何も得られない




$$
\begin{array}{c c}
 & \begin{array}{cccc} \mathrm{A} & \mathrm{B} & \mathrm{C} & \mathrm{D} \end{array} \\
\begin{array}{c} \mathrm{A} \\ \mathrm{B} \\ \mathrm{C} \\ \mathrm{D} \end{array} &
\begin{bmatrix}
0 & 1/2 & 0 & 0 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 0 & 0 & 1/2 \\
1/3 & 1/2 & 0 & 0
\end{bmatrix}
\end{array} = \mathbf{M}
$$

# 行き止まり（Dead Ends）

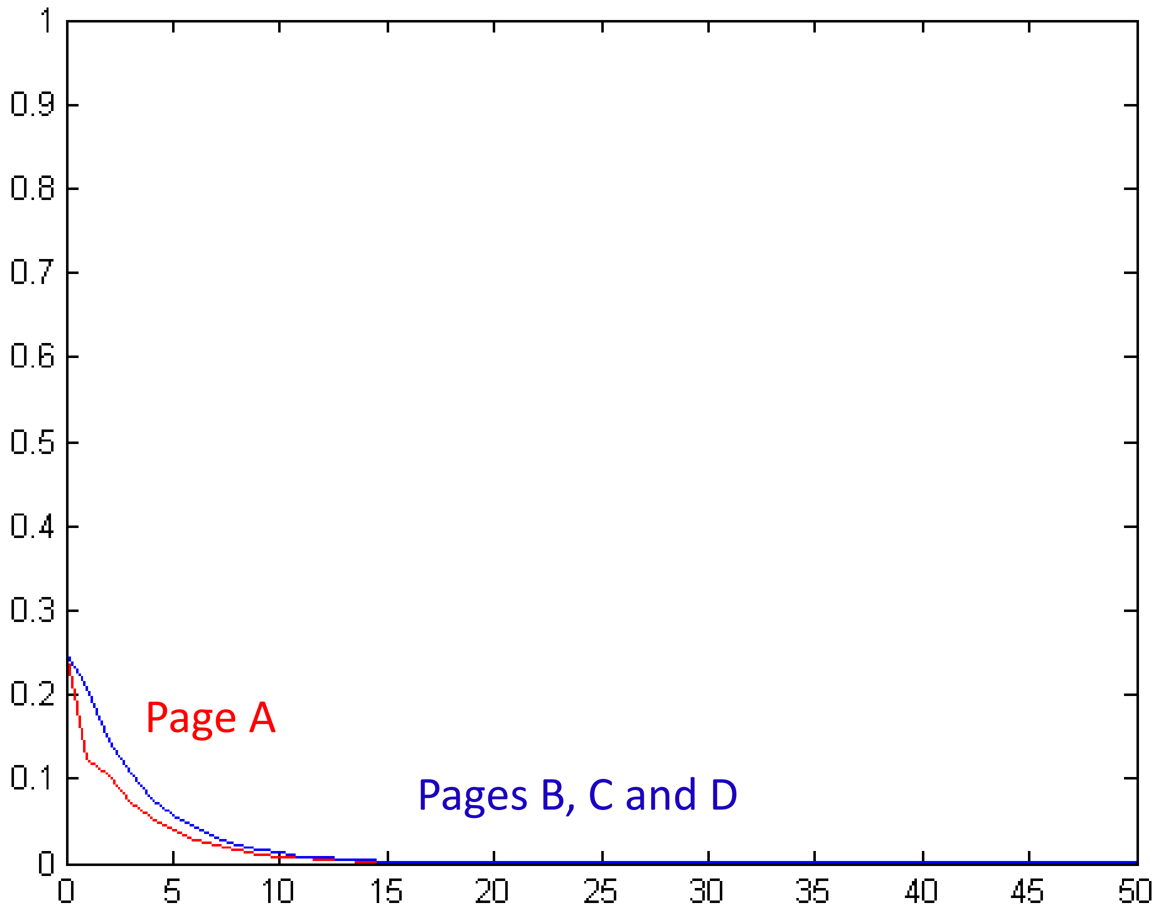
$$\mathbf{v}_0 = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}^T \qquad \sum \mathbf{v}_0 = 1$$

$$\mathbf{v}_1 = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 3/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix} \qquad \sum \mathbf{v}_1 = 18/24 = 0.75$$

$$\mathbf{v}_2 = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 3/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix} = \begin{bmatrix} 5/48 \\ 7/48 \\ 7/48 \\ 7/48 \end{bmatrix} \qquad \sum \mathbf{v}_2 = 26/48 = 0.54$$

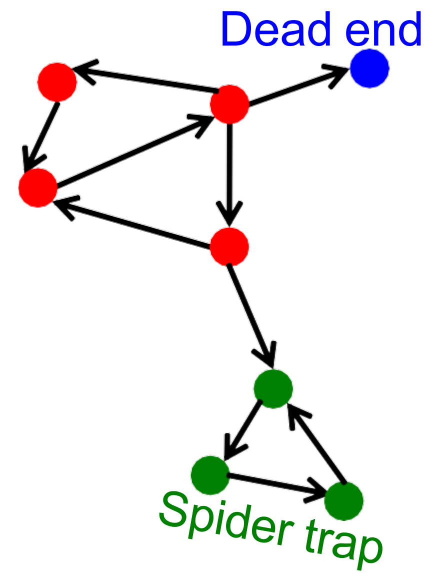
$$\mathbf{v}_3 = \begin{bmatrix} 21/288 & 31/288 & 31/288 & 31/288 \end{bmatrix}^T \qquad \sum \mathbf{v}_3 = 114/288 = 0.40$$

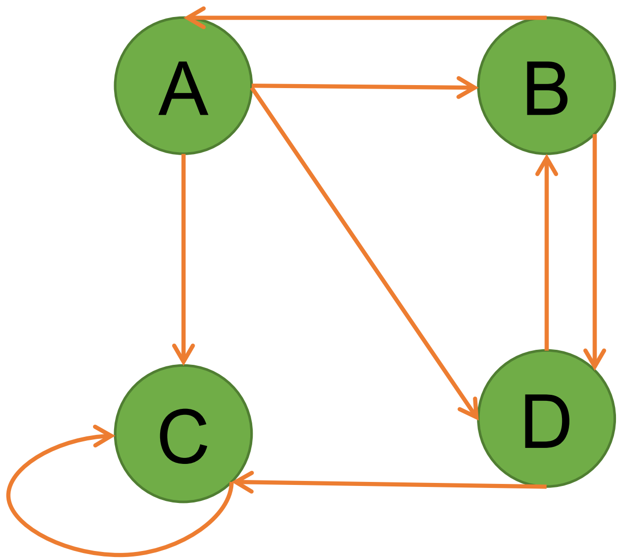
# 行き止まり（Dead Ends）



# スパイダートラップ(Spider Trap)

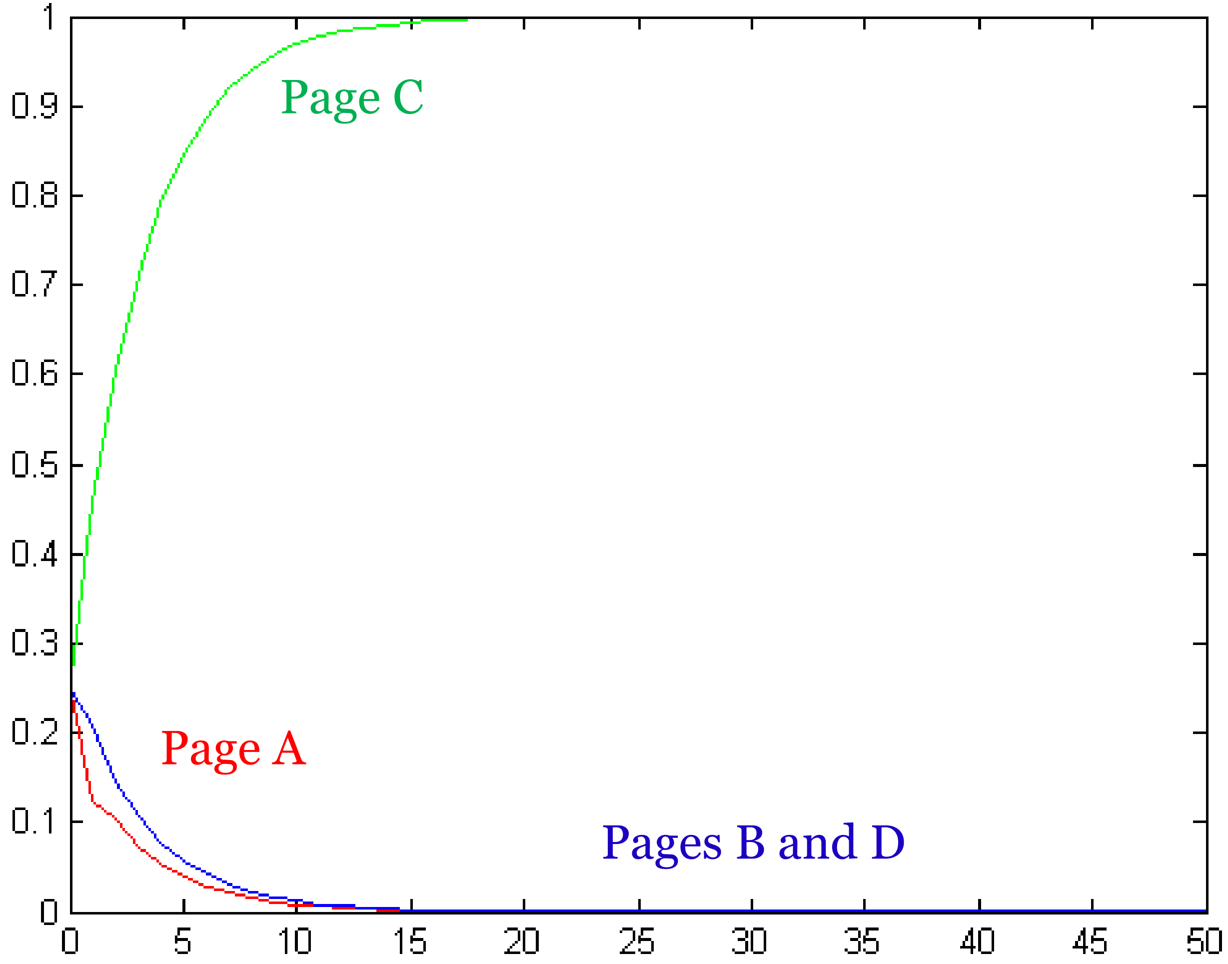
- スパイダートラップ　そのページ群のすべてのページは出リンクを持っているが，そのページ群以外のどの他のページにはリンクしていない
- ウェブの世界で意図的にか，意図せずに現れうる
- スパイダートラップによって，PageRankの計算が，すべてのPageRankをこのスパイダートラップ内に置くように仕向けられる原因となる



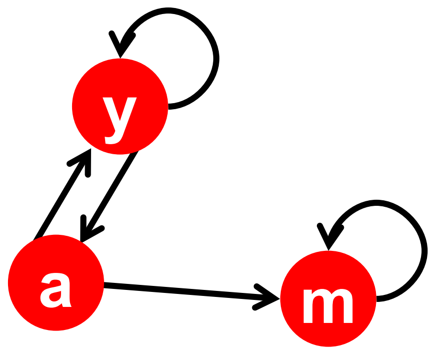

$$\begin{array}{c} \\ A \\ B \\ C \\ D \end{array}
\begin{array}{c}
\begin{array}{cccc} A & B & C & D \end{array} \\
\begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}
\end{array} = \mathbf{M}$$

# スパイダートラップ(Spider Trap)



# スパイダートラップ(Spider Trap)

## 練習




m is a spider trap

| | y | a | m |
|---|---|---|---|
| y | ½ | ½ | 0 |
| a | ½ | 0 | 0 |
| m | 0 | ½ | 1 |

$$\mathbf{v_y} = \mathbf{v_y}/2 + \mathbf{v_a}/2$$

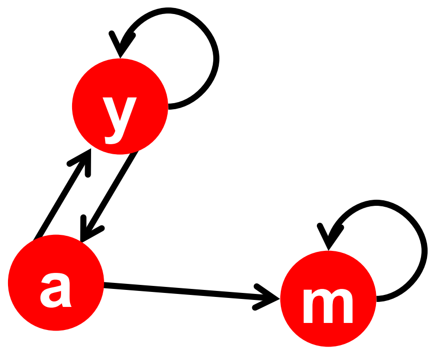
$$\mathbf{v_a} = \mathbf{v_y}/2$$

$$\mathbf{v_m} = \mathbf{v_a}/2 + \mathbf{v_m}$$

Iteration 0, 1, 2, …

# スパイダートラップ(Spider Trap)

## 練習




m is a spider trap

| | y | a | m |
|---|---|---|---|
| y | ½ | ½ | 0 |
| a | ½ | 0 | 0 |
| m | 0 | ½ | 1 |

$$\mathbf{v_y} = \mathbf{v_y}/2 + \mathbf{v_a}/2$$
$$\mathbf{v_a} = \mathbf{v_y}/2$$
$$\mathbf{v_m} = \mathbf{v_a}/2 + \mathbf{v_m}$$

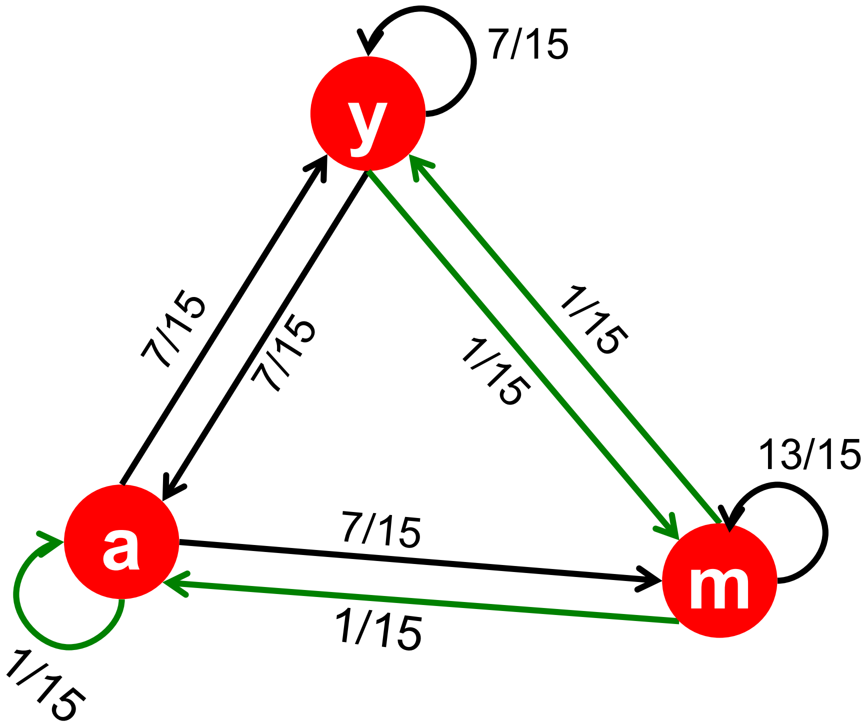
$$\begin{bmatrix} v_y \\ v_a \\ v_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{matrix}$$

# テレポートで問題解決

- ランダムサーファーが現在のページから出リンクをたどるかわりに，小さな確率でランダムなページにテレポート(teleport) することを許す

$$\mathbf{v}_i = \beta \mathbf{M} \mathbf{v}_{i-1} + (1-\beta)\mathbf{v}_0$$

# テレポートで問題解決




$$0.8\ \underbrace{\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}}_{\mathbf{M}} + 0.2\ \underbrace{\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}}_{[1/N]_{NxN}}$$

$$\mathbf{A} = \begin{array}{c} y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array} \quad \begin{array}{c} 0.33 \\ 0.20 \\ 0.46 \end{array} \quad \begin{array}{c} 0.24 \\ 0.20 \\ 0.52 \end{array} \quad \begin{array}{c} 0.26 \\ 0.18 \\ 0.56 \end{array} \quad \ldots \quad \begin{array}{c} 7/33 \\ 5/33 \\ 21/33 \end{array}$$

# テレポートで問題解決

## 行列とベクトルの乗算

$$\boldsymbol{v}^{new} = \boldsymbol{A} \cdot \boldsymbol{v}^{old}$$

## 例 N = 1 billion pages

- We need 4 bytes for each entry
- 2 billion entries for vectors, approx 8GB
- Matrix **A** has $N^2$ entries
  - $10^{18}$ is a large number!

$$\mathbf{A} = \beta \cdot \mathbf{M} + (1-\beta)\,[1/N]_{N \times N}$$

$$\boldsymbol{A} = 0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$= \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

# テレポートで問題解決

- $\mathbf{v} = \boldsymbol{A} \cdot \mathbf{v}$, **where** $\boldsymbol{A_{ij}} = \boldsymbol{\beta}\ \boldsymbol{M_{ij}} + \frac{\mathbf{1-\boldsymbol{\beta}}}{\boldsymbol{N}}$
- $v_i = \sum_{j=1}^{N} A_{ij} \cdot v_j$
- $v_i = \sum_{j=1}^{N} \left[\beta\ M_{ij} + \frac{1-\beta}{N}\right] \cdot v_j$

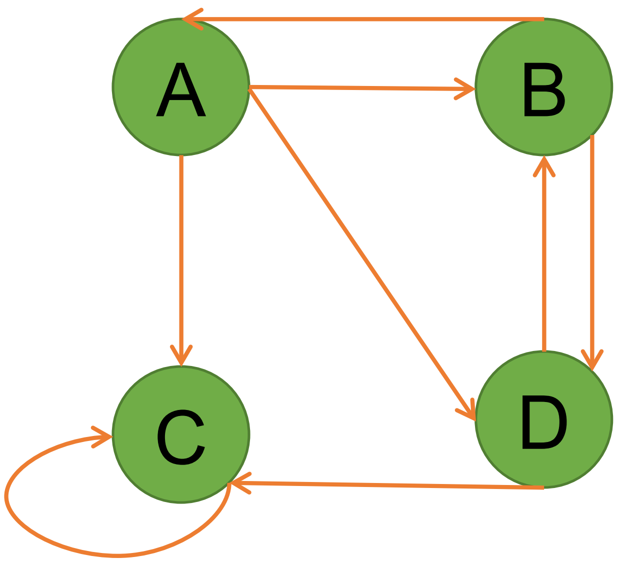
$$= \sum_{j=1}^{N} \beta\ M_{ij} \cdot v_j + \frac{1-\beta}{N} \sum_{j=1}^{N} v_j$$

$$= \sum_{j=1}^{N} \beta\ M_{ij} \cdot v_j + \frac{1-\beta}{N}$$ since $\sum v_j = 1$

- **So we get:** $\boldsymbol{v} = \boldsymbol{\beta}\ \boldsymbol{M} \cdot \boldsymbol{v} + \left[\frac{\mathbf{1-\boldsymbol{\beta}}}{\boldsymbol{N}}\right]_{\boldsymbol{N}}$

$$\boxed{\mathbf{v}_i = \beta \mathbf{M} \mathbf{v}_{i-1} + (1-\beta)\mathbf{v}_0}$$

# テレポートで問題解決




$$\beta = 0.8$$

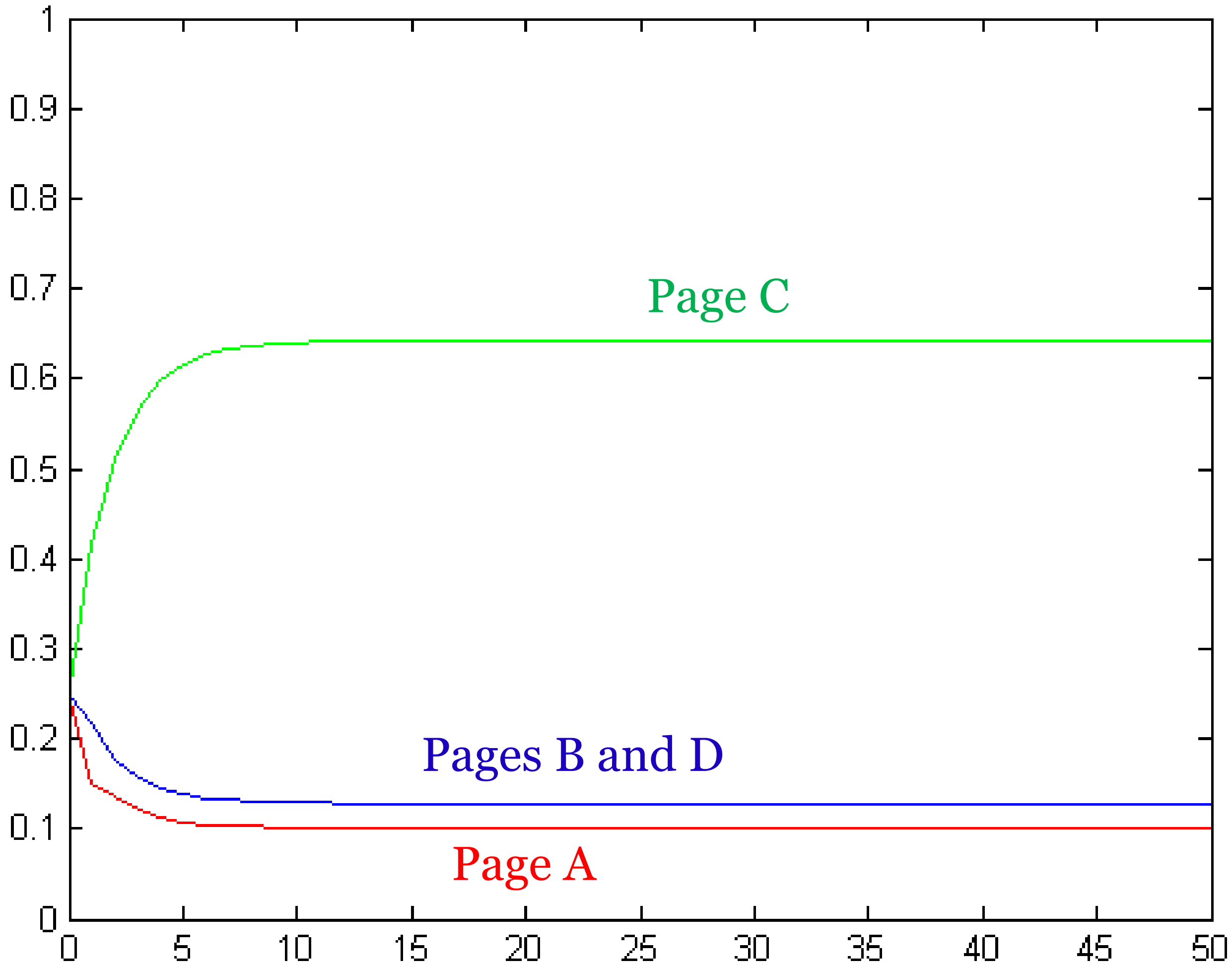
$$\begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

$$\mathbf{v}_0 = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}^T$$

$$\mathbf{v}_1 = 0.8 \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} + 0.2 \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 9/60 \\ 13/60 \\ 25/60 \\ 13/60 \end{bmatrix}$$

$$\mathbf{v}_2 = 0.8 \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 1 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 9/60 \\ 13/60 \\ 25/60 \\ 13/60 \end{bmatrix} + 0.2 \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 41/300 \\ 53/300 \\ 153/300 \\ 53/300 \end{bmatrix}$$

# スパイダートラップ(Spider Trap)



THE END