問題を見ずに問題を解く
～ 定数時間アルゴリズムとは？ ～

吉田　悠一
国立情報学研究所
2014年1月22日

自己紹介

- 吉田悠一
- 専門：理論計算機科学
  - 計算とは何かを考える学問
  - P vs NP、乱数、暗号
- 出身：大阪府枚方市
- 洛南高校

  ⇒ 京都大学

  ⇒ (Preferred Infrastructure, Inc.)

  ⇒ 国立情報学研究所
- 4日後に結婚式！！

あらすじ

- ソーシャルグラフ: 六次の隔たり

- 六次の隔たりをどうやって判定するか？
  - 古典的な判定方法
  - 定数時間アルゴリズムとは
  - 定数時間アルゴリズムによる判定方法

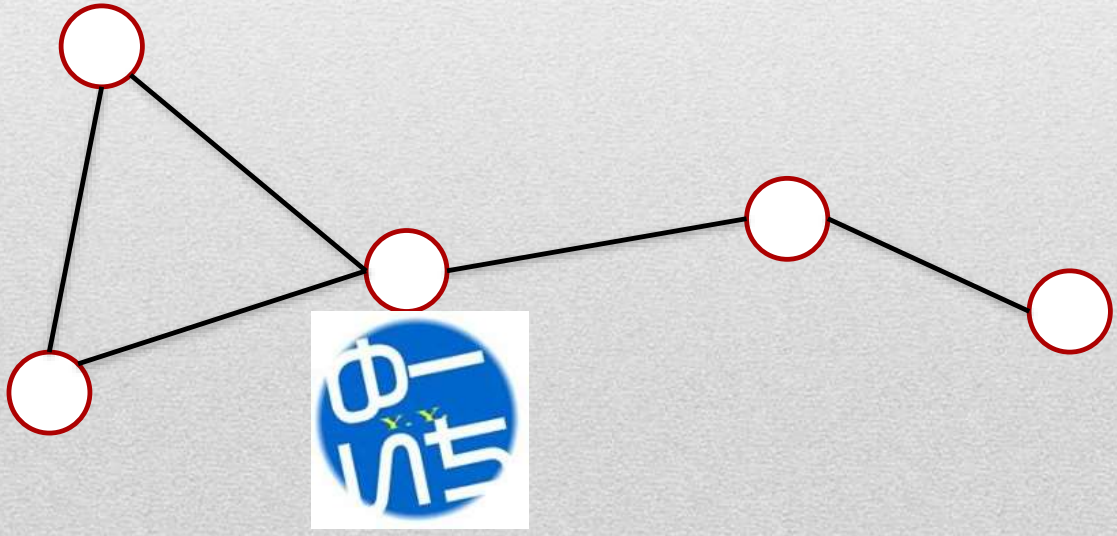
- 定数時間アルゴリズムのその他の応用
- 私の最近の研究

ソーシャルグラフ

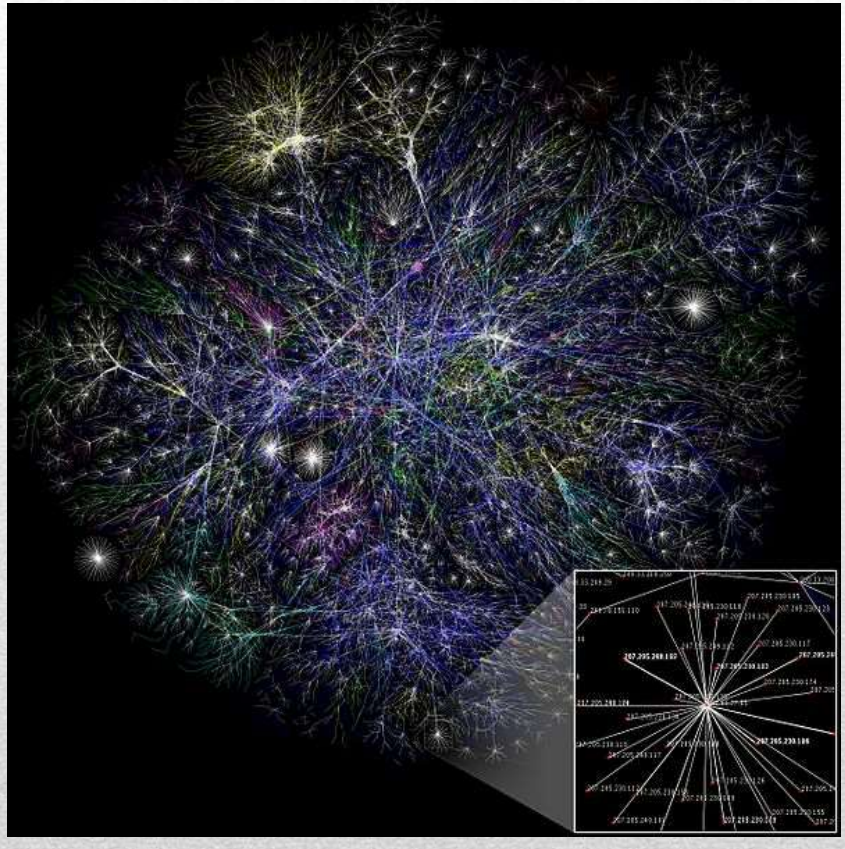
好きなSNSを考える(Facebook, Mixiなど)

以下の様にグラフを作る

- ユーザ ⇒ 点
- 友人関係 ⇒ 点と点を結ぶ枝

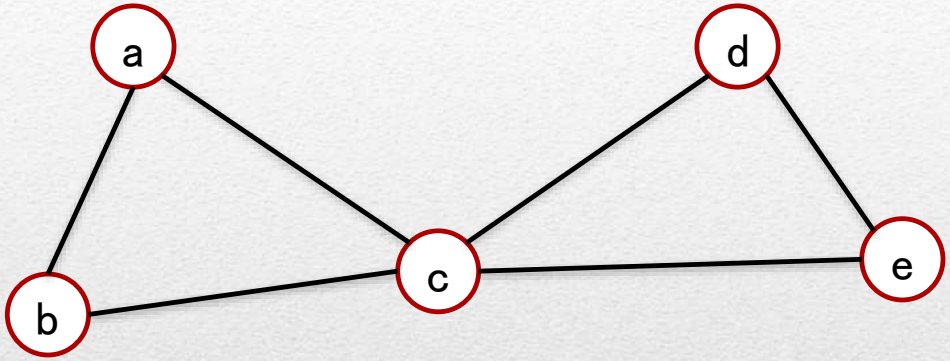


ソーシャルグラフ



http://en.wikipedia.org/wiki/Network_mapping

グラフの用語

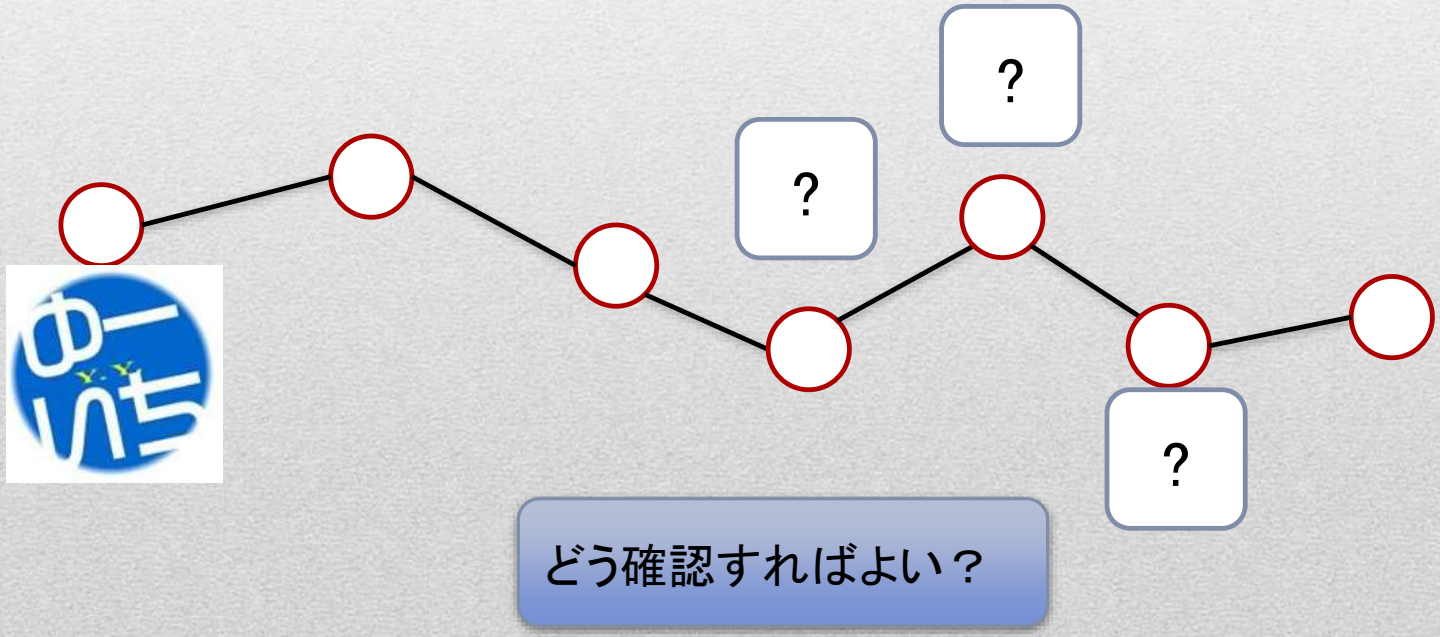


- 頂点数 = 5
- 枝数 = 5
- 頂点aと頂点cの距離 = 1
- 頂点aと頂点eの距離 = 2
- グラフの直径 = 2

六次の隔たり

友人関係を高々6回たどることで、(殆ど)全ての人に到達することができる。

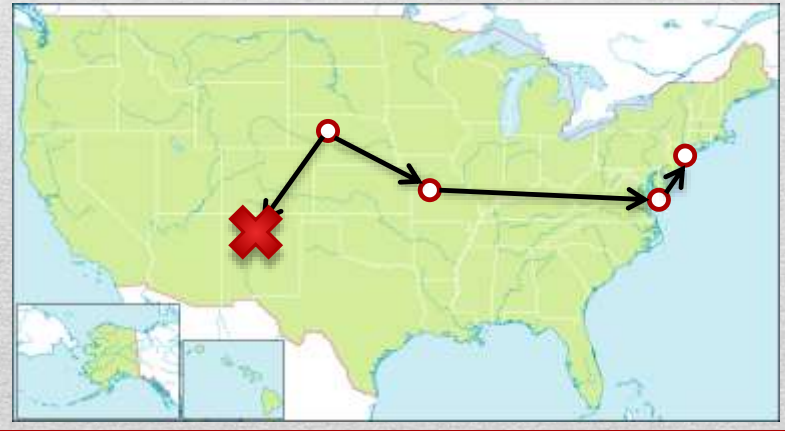
⇔ 直径が(ほぼ)6



ミルグラムの実験（1967）

- アメリカ・ネブラスカ州の160人に以下の手紙を送った。

  「同封写真はボストン在住の株式仲介人です。本人を知っていたらこの手紙を転送してください。そうでなければ、知っていそうな人にこの手紙を転送してください。」



ミルグラムの実験

- 42通が実際に届いた。
- 途中で経由した人数の平均値は5.83人。

六次の隔たり！？

余り鵜呑みにはできない。

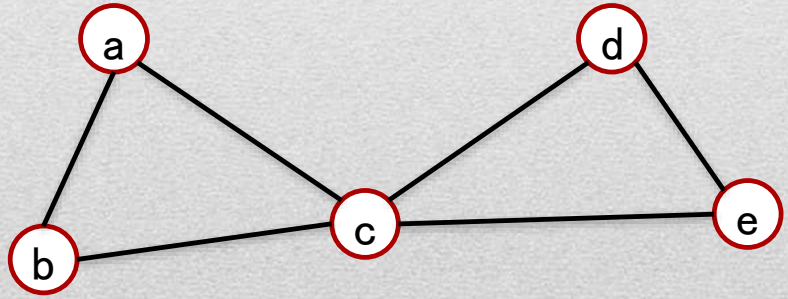
- 25%しか届いていない
  - 残りは遠いから届かなかったのでは？
- ボストンは都会なので辿り着きやすいかもしれない
- 追試も出来ない

SNS企業による実験

現在はSNSを通じてソーシャルグラフ全体が手に入るので、もう少し正確な実験が可能。

平均距離（≠ 直径）

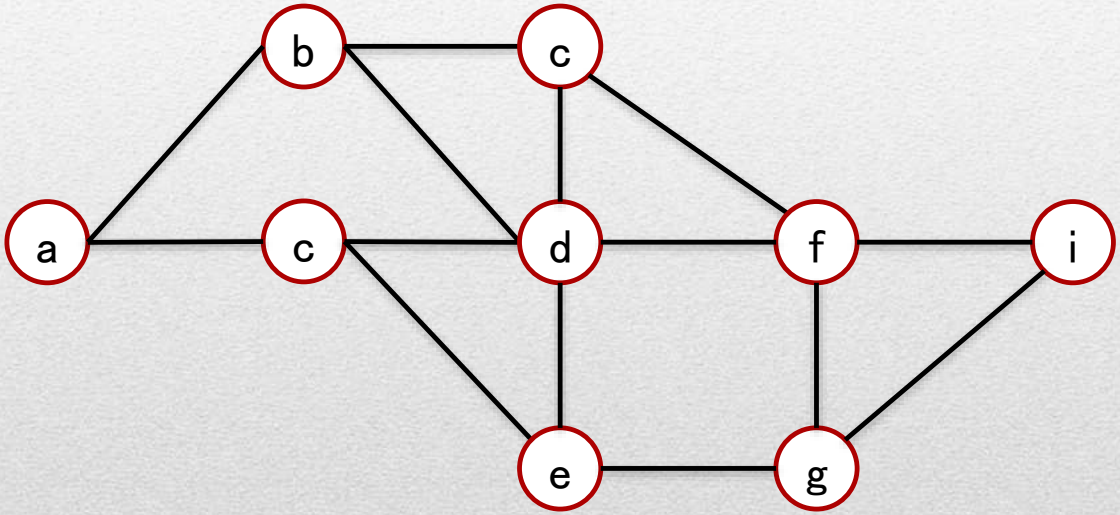
- Facebook: 4.74 (2011年時点)
- Twitter: 4.67 (2010年時点)



何故直径を発表しない？

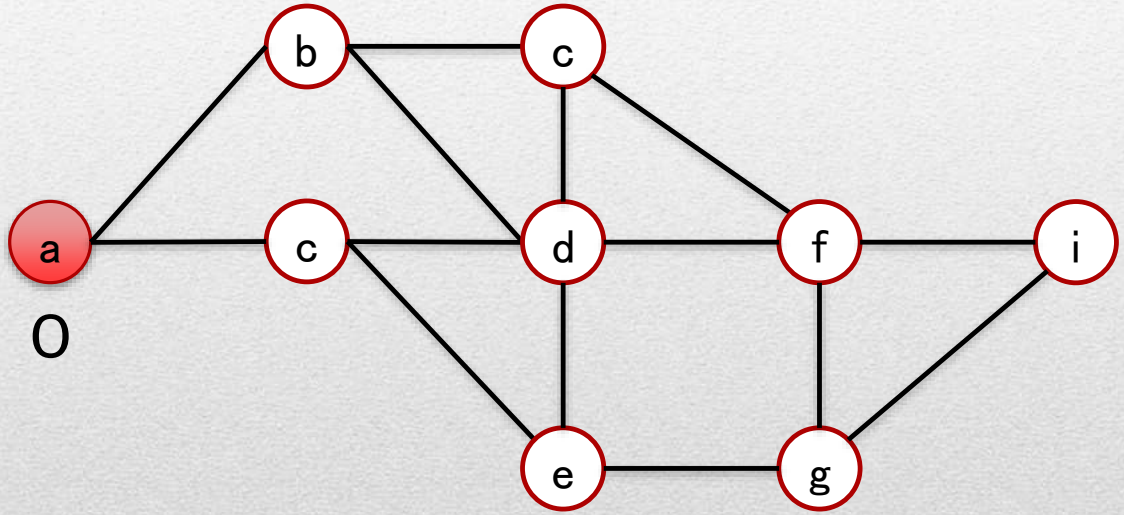
幅優先探索

一点 vs 全点の距離を求める



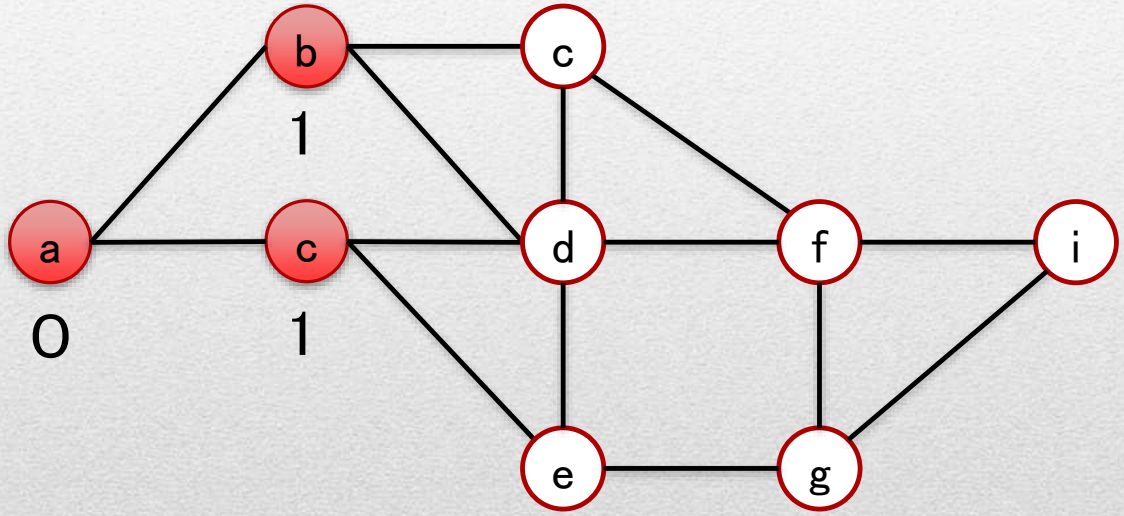
幅優先探索

一点 vs 全点の距離を求める



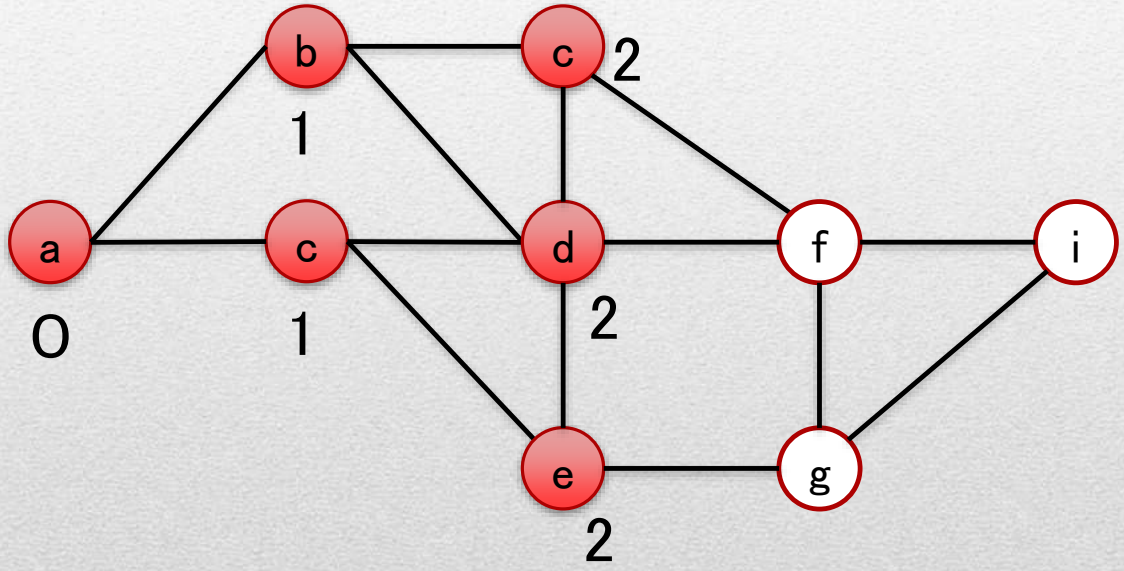
幅優先探索

一点 vs 全点の距離を求める



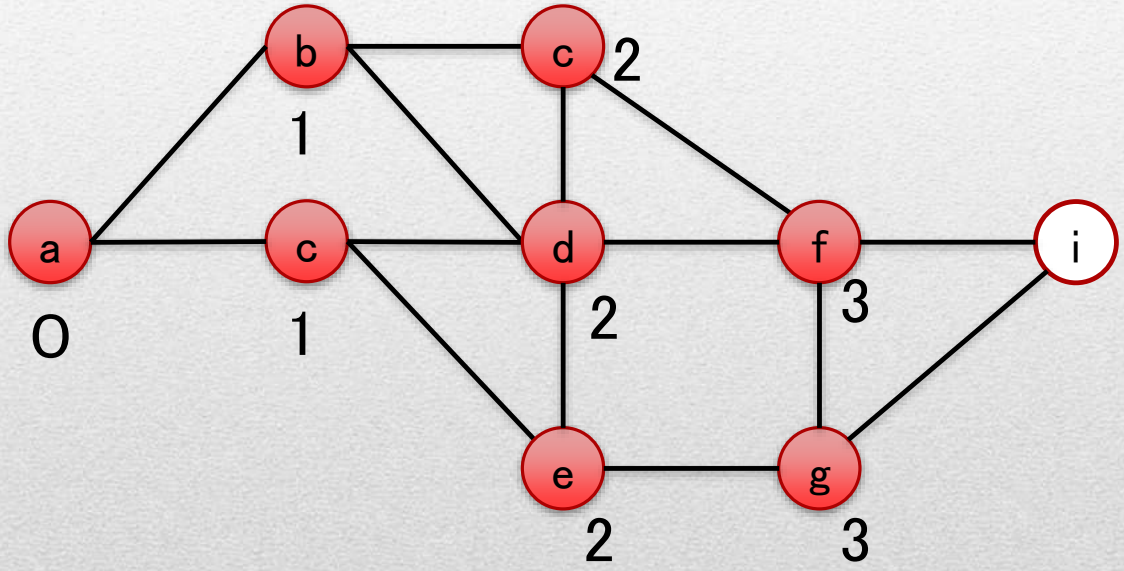
幅優先探索

一点 vs 全点の距離を求める



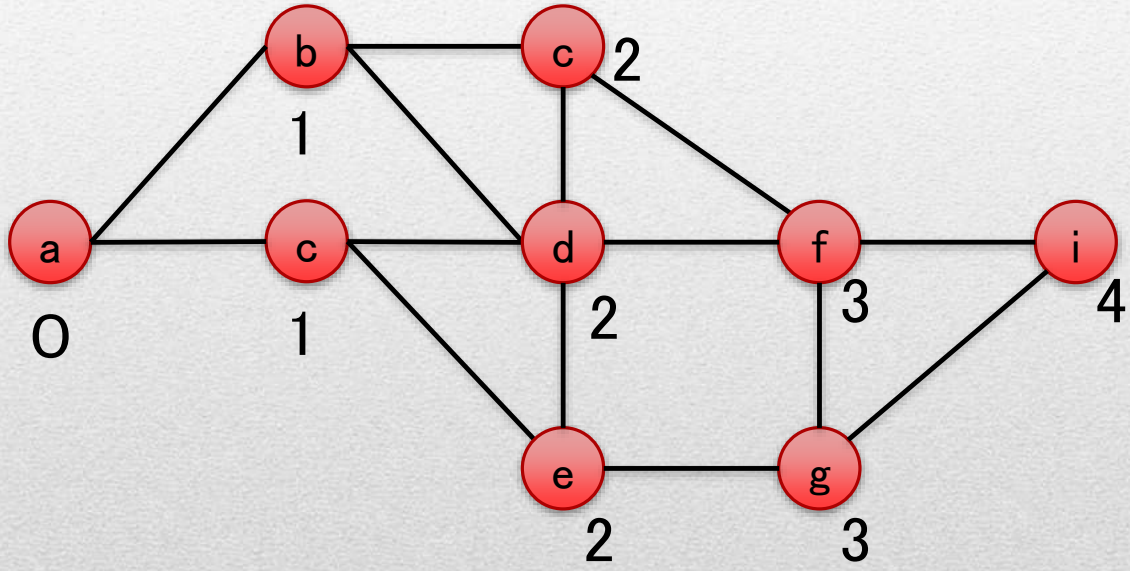
幅優先探索

一点 vs 全点の距離を求める



幅優先探索

一点 vs 全点の距離を求める



計算時間 ≈ 枝数

直径の求め方

- 全ての頂点から幅優先探索する。
- 全点 vs 全点の距離が手に入る。

	a	b	c	. . .	i
a	0	1	1		4
b	1	0	2		3
c	1	2	0		3
:					:
i	4	3	3	. . .	0

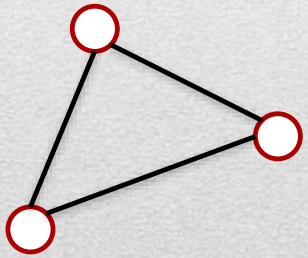
- 一番大きい距離が直径

全体の計算時間

直径の計算に必要な計算ステップ数 ≈ 頂点数 × 枝数

最近のFacebook

- ユーザ数: 10億人 ⇒ 頂点数10億
- 一人あたりの平均友人数: 130 ⇒ 枝数650億



最近のコンピュータの速度は大体

計算ステップ数＝1億 ⇒ 1秒

全体の計算時間

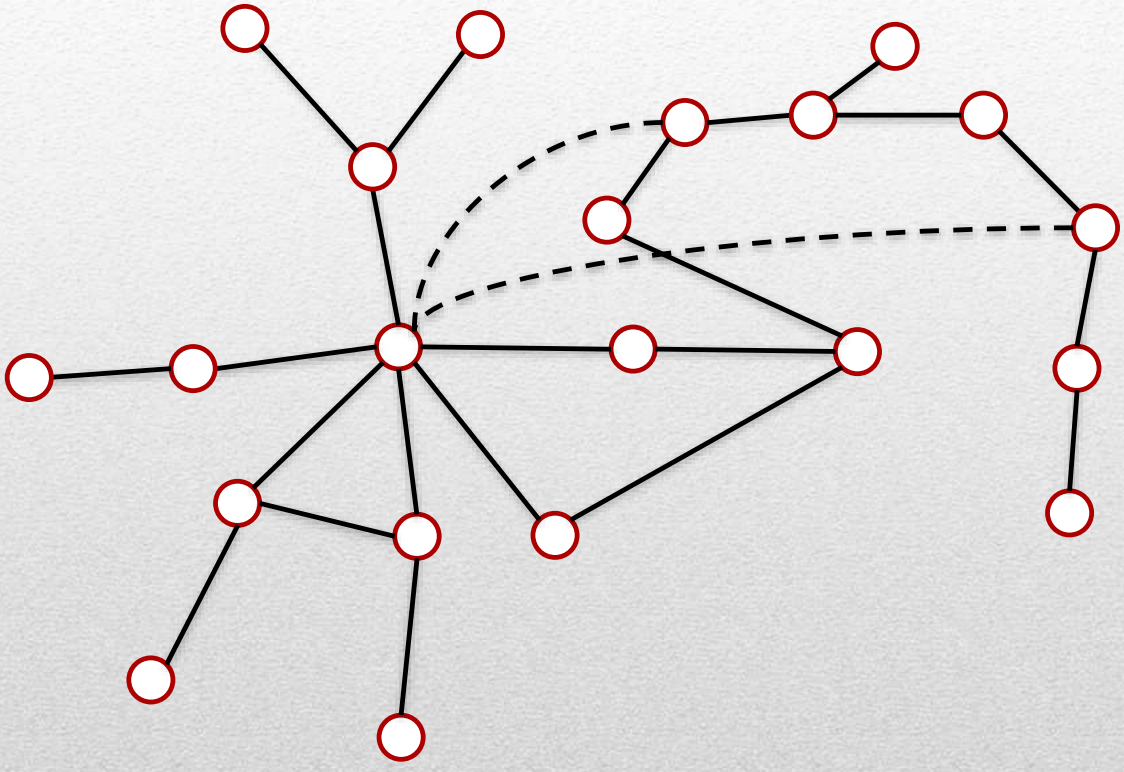
計算ステップ数 = 頂点数 × 枝数 = 6500京 ⇒ 2万年

遅すぎて使い物にならない！

少しぐらい間違っても良いので、
劇的に速く出来ないか？

定数時間アルゴリズム

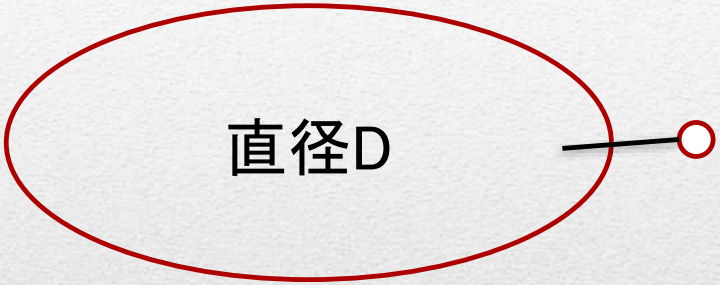
考察: 直径は不安定なパラメータ



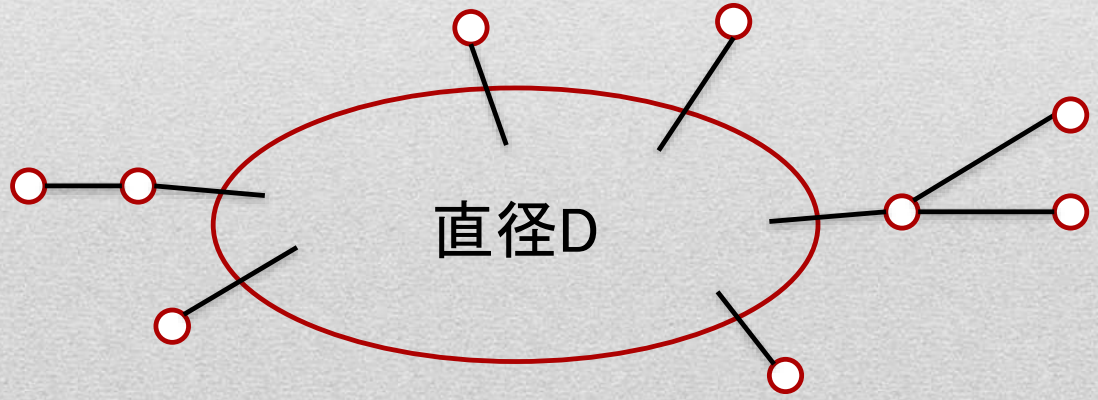
直径は11 だが、殆ど4みたいなもの。

考察: 直径は不安定なパラメータ

- 「直径高々Dなグラフ」に近い



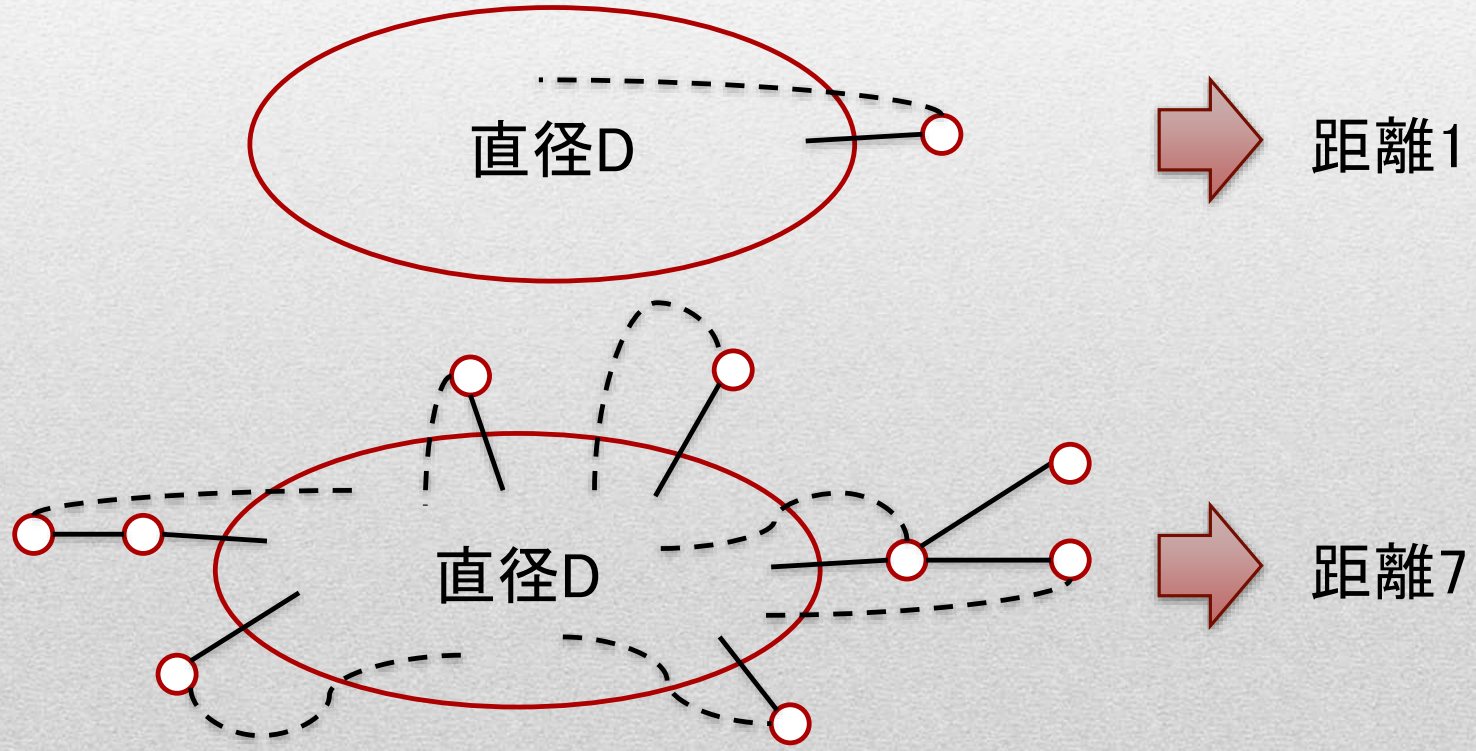
- 「直径高々Dなグラフ」から遠い



「直径高々D」との距離

直径高々Dとの距離:

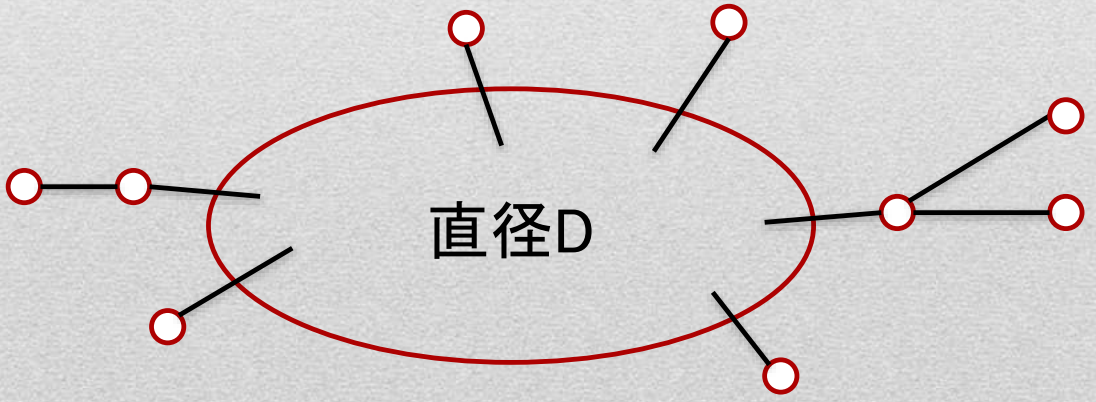
直径をD以下にするのに足すべき枝の本数



性質検査の考え方

- 直径高々Dなグラフ

- 直径高々Dなグラフから遠い



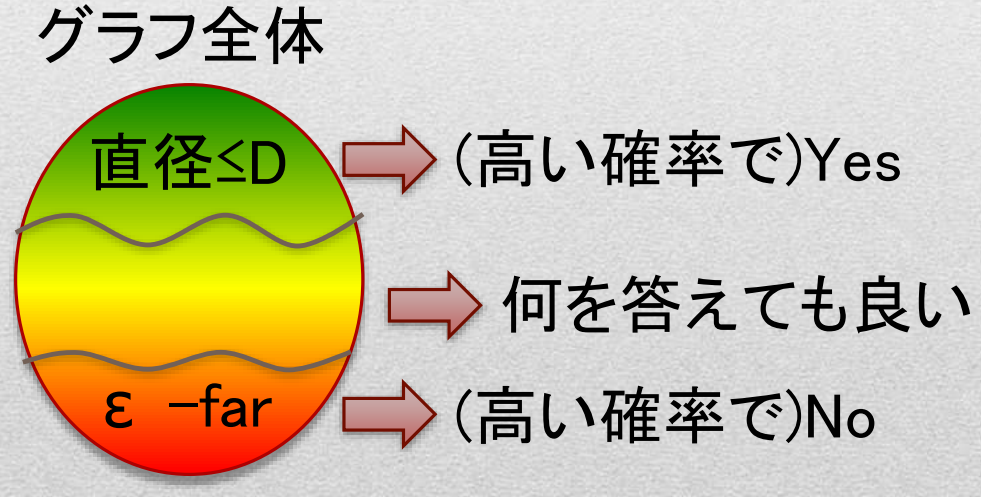
この二つを
区別することに
しよう

性質検査

直径≤Dからε −far:

直径≤Dなグラフとの距離が ε ×(頂点数) 以上。

直径≤Dという性質に対する検査アルゴリズム:



性質検査の狙い

Dを正確に求めるには(頂点数)×(枝数)ステップ必要

⇒グラフが大きくなればなるほど遅くなる

⇒グラフのε 割合の誤差を許すことで速くなるかも？

多くの検査アルゴリズムは

グラフのサイズに依らない時間（定数時間）

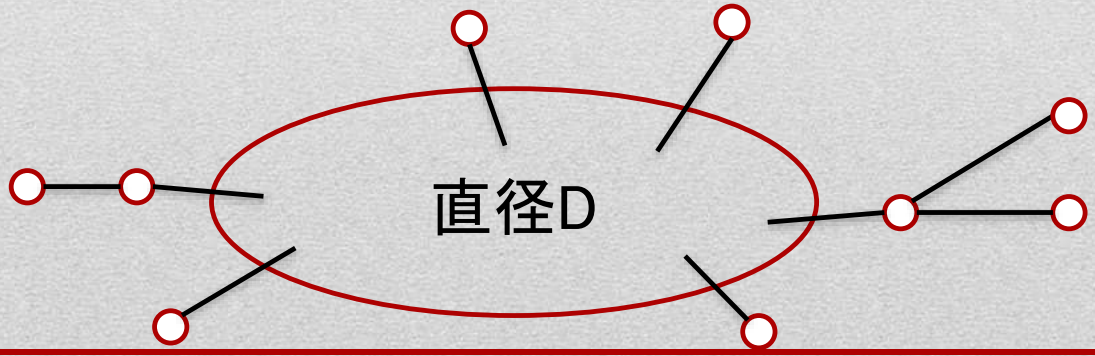
で動作する

注: ε =0.01などに固定

何故定数時間になる？

- 直径≤Dと区別つきにくいが、ε −farではないので何を出力してもよい。

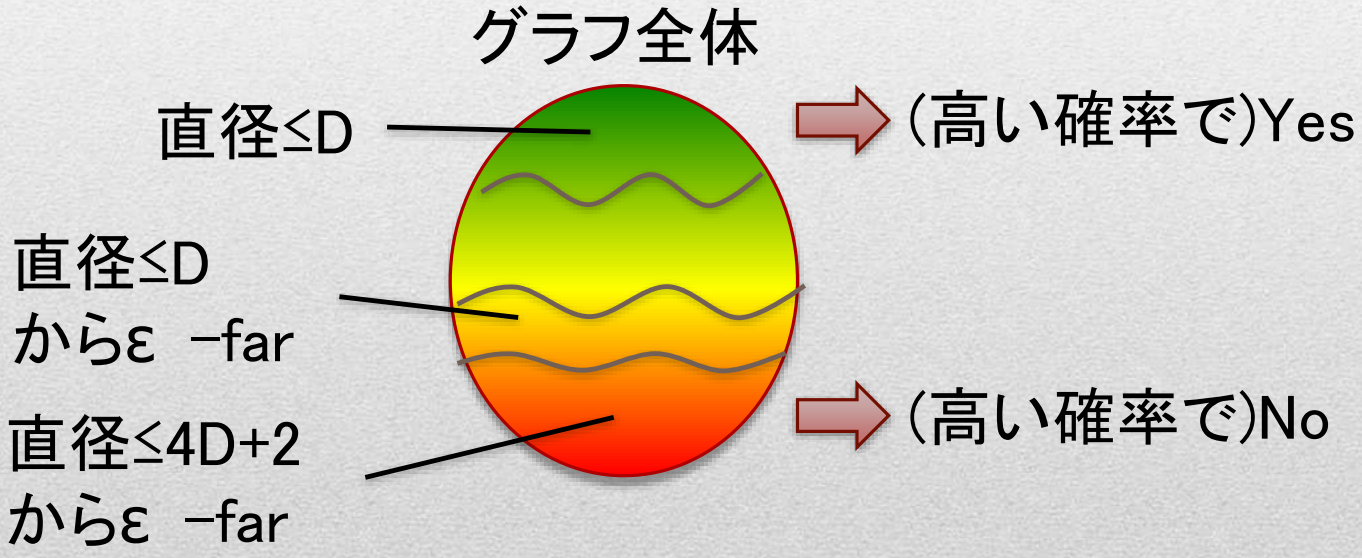
- 直径≤Dからε −farなのでNoと出力しないといけない。しかし、端の頂点は簡単に見つかる。



更に問題を緩める

直径≤Dと直径≤Dからε −farを区別するのは難しい。

⇒ 直径≤Dと直径≤4D+2からε −farを区別する。

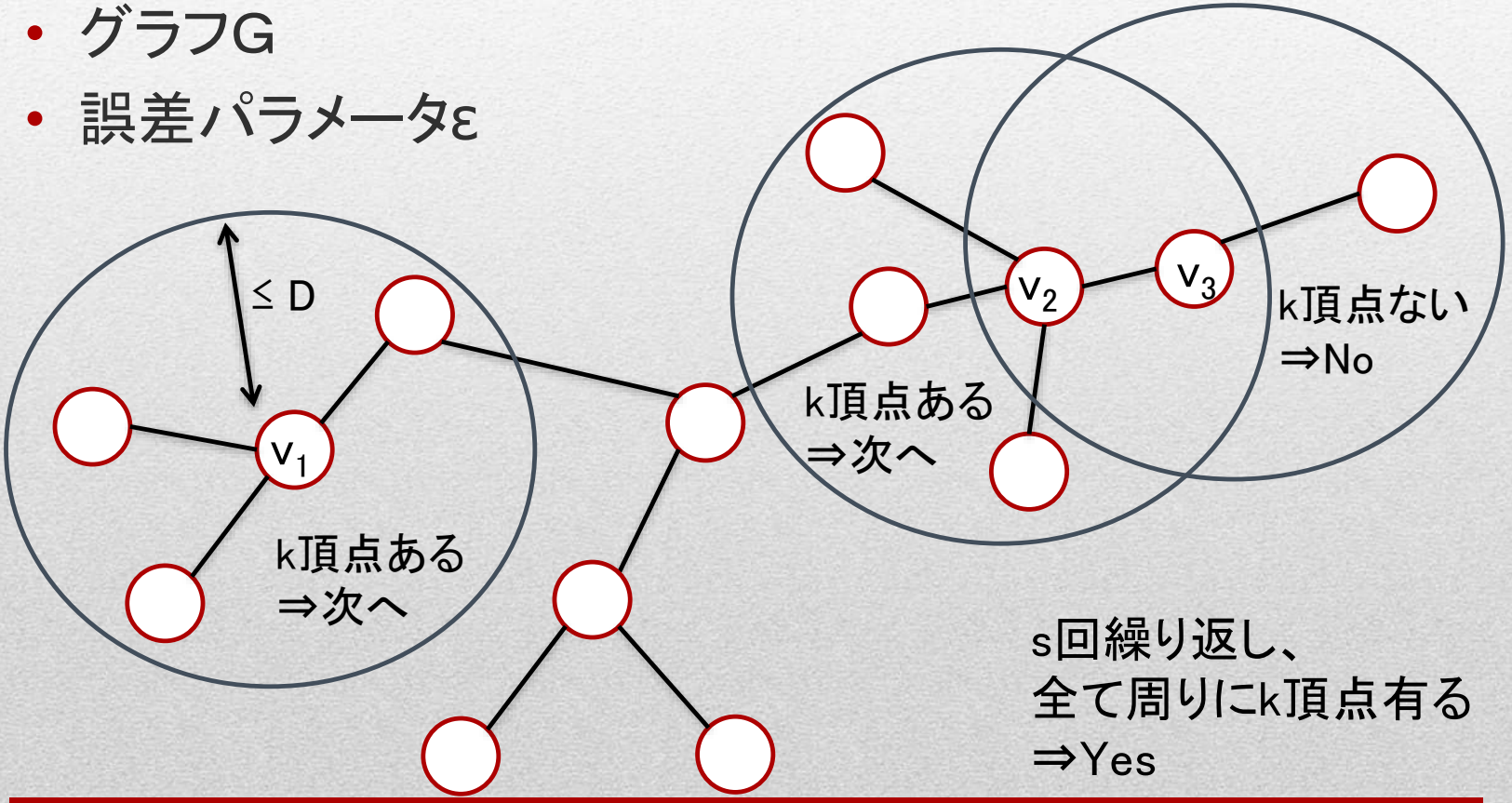


検査アルゴリズム概要

入力:

- グラフG
- 誤差パラメータε

パラメータkとsを適切に設定



s回繰り返し、
全て周りにk頂点有る
⇒Yes

検査アルゴリズム

入力:

- グラフG
- 誤差パラメータε

以下の様に設定

$s = 6/\varepsilon$

$k = 2/\varepsilon$

アルゴリズム:

頂点$v_1,\cdots,v_s$をランダムに選ぶ。

for i = 1 to s

　v_iから半径D以内をk頂点だけ幅優先探索

　if k頂点みつからなかった then Noと出力。

Yesと出力。

アルゴリズムの計算時間の解析

計算ステップ数 ≈ $sk = 12/\varepsilon^2$

グラフのサイズに依らない＝定数時間

Facebookの場合:

- $\varepsilon = 0.01$

  ⇒ 計算ステップ数 = 120000

  ⇒ 1.2ミリ秒
- $\varepsilon = 0.001$ ⇒ 120ミリ秒
- $\varepsilon = 0.0001$ ⇒ 12秒

アルゴリズムの正しさの解析

- 良い頂点: 距離D以内にk頂点以上存在
- 悪い頂点: 距離D以内にk頂点未満しか存在しない。
  (注: 悪い頂点が一つでも見つかるとNoと出力)

以下を示す

- 直径≤Dの時 ⇒ 全ての頂点が良い
- 直径≤4D+2からε −farの時 ⇒ 多くの頂点が悪い

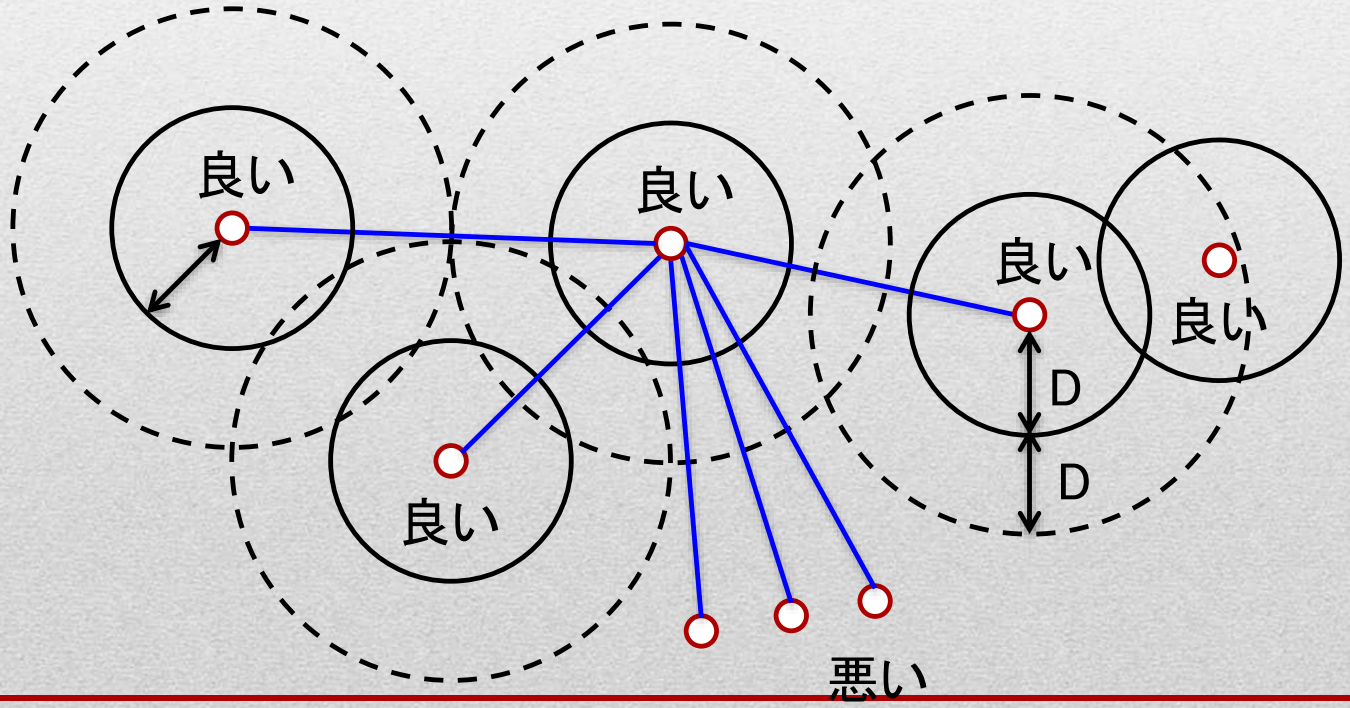
直径≤Dの時

どの頂点の周りにも距離D以内にn頂点存在する。

⇒全ての頂点が良い

直径≦4D+2からε −farの時

[定理] 悪い頂点がt個ある時、t+n/k本の枝を
追加することで半径4D+2に出来る。



直径≤4D+2からε −farの時

ε −far ⇒ 少なくともε n本枝を足さないといけない。

⇒ $t + n/k \geq \varepsilon n$

⇒ $t \geq \varepsilon n/2$ ($k=2/\varepsilon$ より)

一度も悪い頂点を選ばない確率:

$(1 - (\varepsilon n/2)/n)^s = (1-\varepsilon/2)^s$

$\leq (1-\varepsilon/2)^{6/\varepsilon}$

$\leq e^{-3}$ ($1 - x \leq e^{-x}$より)

≤ 0.05

⇒ 95%以上の確率でNoと出力する。

まとめ

- Facebookの様な巨大なグラフの直径Dを正確に求めると2万年必要。
- 直径≤Dと直径≤4D+2からε –farを区別するので良いのであれば、数秒で済んでしまう。
- しかも後者はグラフの大きさに依らない計算時間。

- 何故定数時間で済むのか？

  許す誤差がグラフの大きさに依存しているから。しかし直径の様な問題の時はむしろ妥当。

- 直径≤Dと直径≤D+4からε –farの区別をすることも可能。

常に同じ大きさの部分だけ見て

問題~~を見ずに~~問題を解く

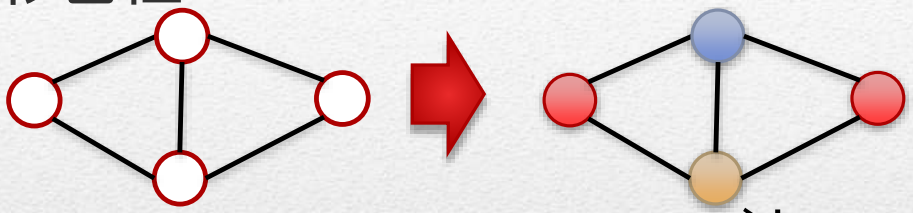
～ 定数時間アルゴリズムとは？ ～

吉田　悠一
国立情報学研究所
2014年1月22日

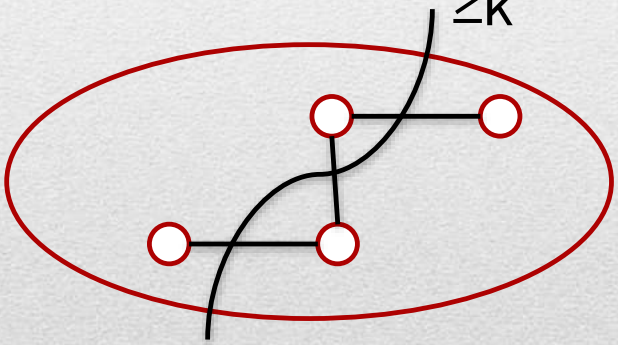
性質検査のその他の応用

その他のグラフの性質に対する検査

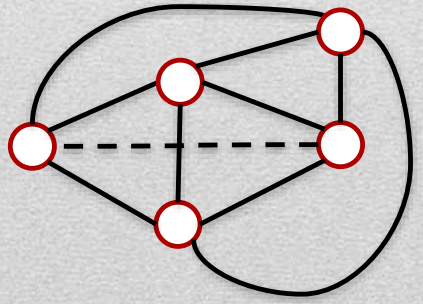
- 三彩色性



- k枝連結性



- 平面性



全て定数時間で
検査可能

性質検査の適用事例

1. 入力データが巨大
2. 入力データは小さいが、問題を解くのが難しい
3. 同じ問題を大量に正確に解きたい

入力データが巨大

例：DNA配列、ソーシャルグラフ、ウェブグラフ

- 全体を読み込むだけで時間がかかる。
  1. 例え完全なデータが有っても数分はかかる
  2. 完全なデータが手に入らないかもしれない

- 性質検査ならば、定数サイズを読むだけで良い
  1. 数分から数秒、さらに短い時間へ
  2. 検査アルゴリズムを走らせながらデータ取得

入力データは小さいが、問題を解くのが難しい

例: 三彩色性

- NP完全問題
- 最速アルゴリズムの計算ステップ数 = 約2^n

  n = 10 ⇒ 0.00001秒

  n = 20 ⇒ 0.01秒

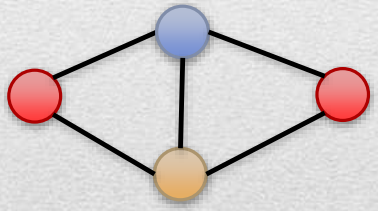
  n = 30 ⇒ 11秒

  n = 40 ⇒ 3時間

  n = 50 ⇒ 130日

  n = 60 ⇒ 365年

  n = 100 ⇒ 400兆年…



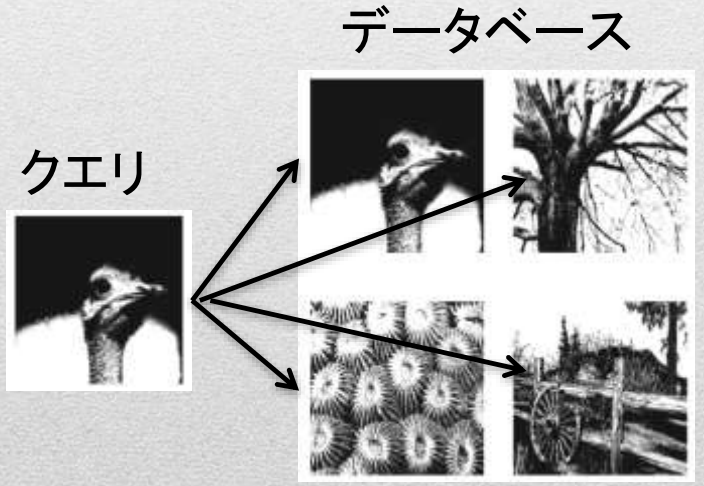
性質検査ならば、定数時間で解ける！

同じ問題を大量に正確に解きたい。

例: 画像検索

性質検査を枝刈りに使う。

farでないものだけ正確に解く。

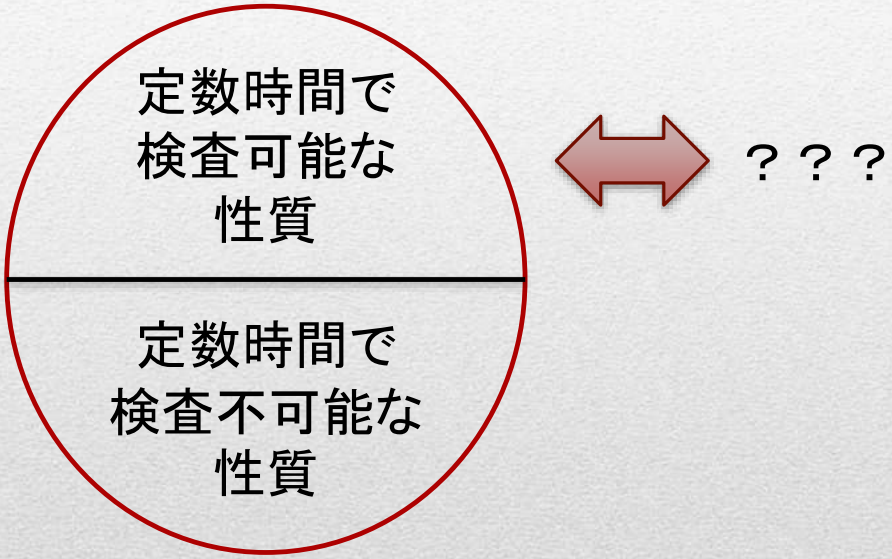


Kleiner et al., “Applying Property Testing to an Image Partitioning Problem,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 256–265.

私の最近の研究内容
（のグラフ版）

性質検査という研究分野の最終目標

- 定数時間で検査可能な性質とは何かを明らかにする

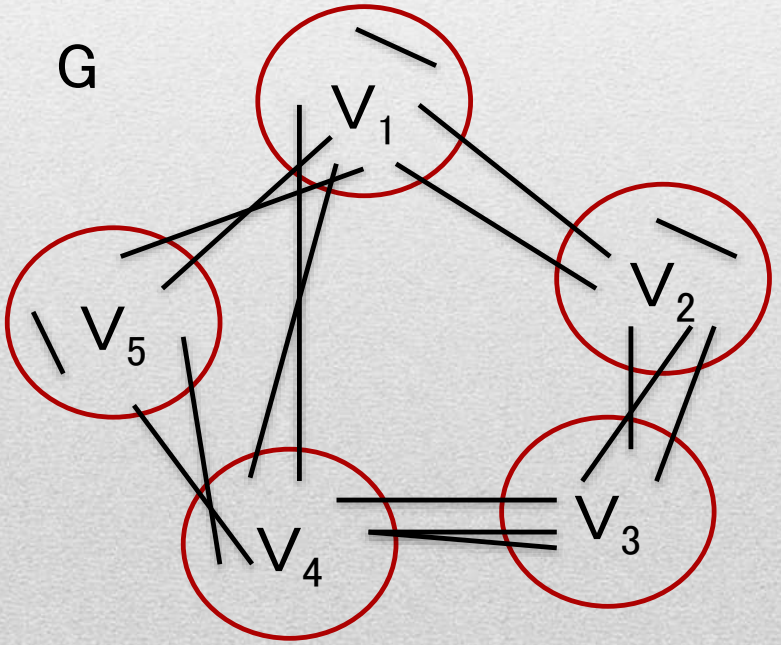


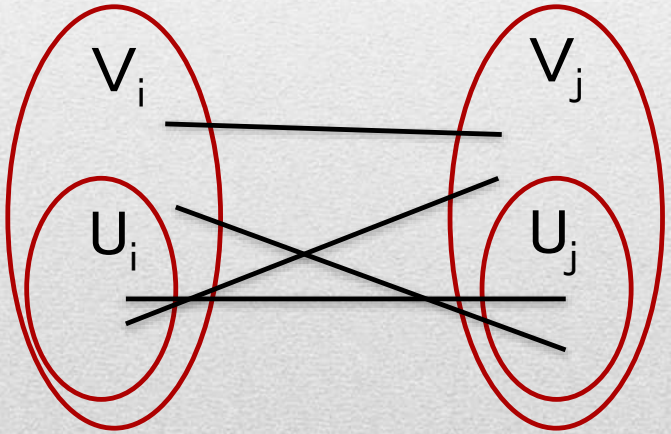
- 局所的な性質と大域的な性質の関係を明らかにする

  例: 悪い頂点の個数と、半径4D+2にするのに足すべき枝数の関係

Szemerédiの正則性補題

任意のグラフは、等しい大きさの部分$V_1,\cdots,V_k$に分けることができ、各ペア(V_i, V_j)は密度η_{ij}のランダムな二部グラフに見える。





V_i–V_j間の枝の本数$\approx \eta_{ij}|V_i||V_j|$

$\Rightarrow$ U_i–U_j間の枝の本数$\approx \eta_{ij}|U_i||U_j|$

定数時間で検査可能な
グラフの性質の特徴付け

性質Pからε −far:

性質Pを満たすには枝をε（頂点数)^{2}本
追加もしくは削除しなければならない。

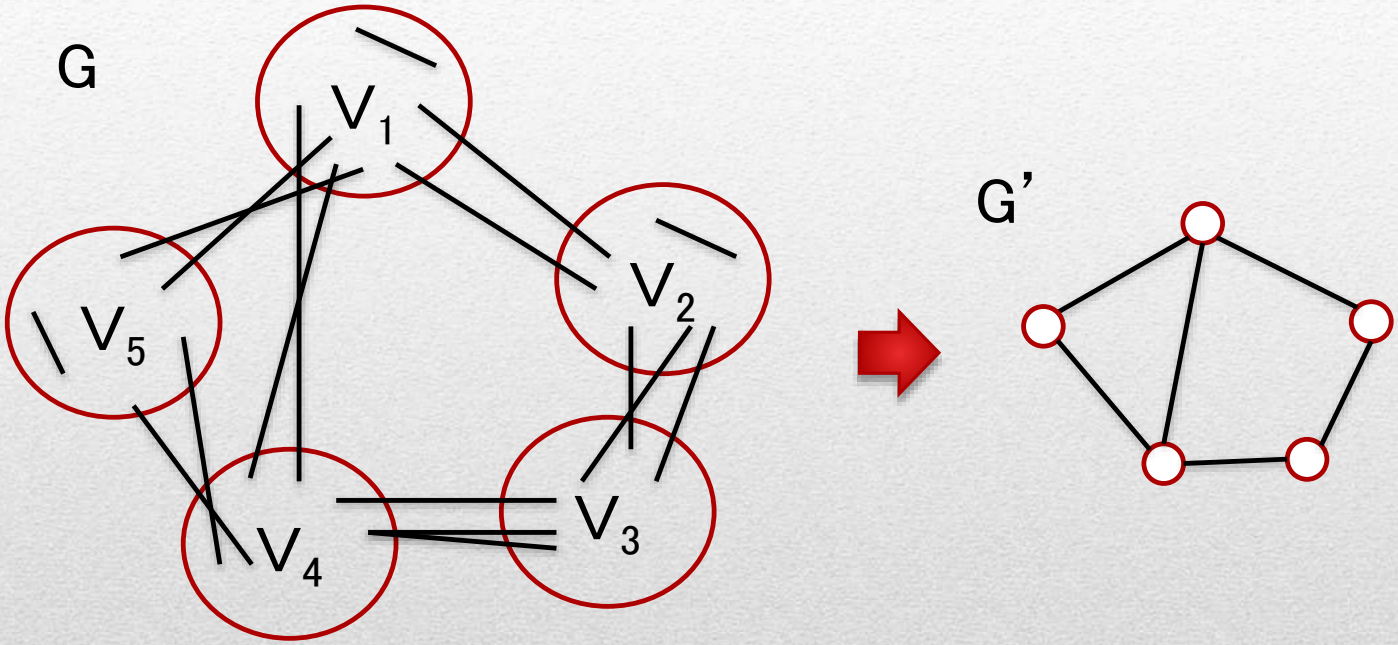
性質Pが定数時間で検査可能

⇔

性質Pが満たされるか否かは、
密度の集合{η_{ij}}だけで決まる

[Alon et al. SICOMP]

三彩色可能性は定数時間で検査可能



Gが三彩色可能かを判定するには、G’が三彩色可能かを調べれば良い。

⇒ 三彩色可能性は定数時間で検査可能。

私の最近の研究

- 関数f: $\mathbb{F}_p^n \to \{0, 1\}$の性質の検査

  例: 線形性、低次の多項式か、有る形に因数分解できるか etc.

- 定数時間で検査可能な必要十分条件を得た
  - 加法的組み合わせ論
  - (高階)調和解析

  などの数学的道具を使用

まとめ

- 問題を正確に解くのではなく、ε 割合の誤差を許すことで、計算時間を定数まで減らすことが出来る。

- 数学的な興味から生まれた分野。
- 現実の応用から着想を得ることで、更に豊かな分野になると期待。

- ご清聴有り難うございました