最大クリーク問題を解くアルゴリズムの新提案

発表者：
谷 研究室
原田英幸

発表内容

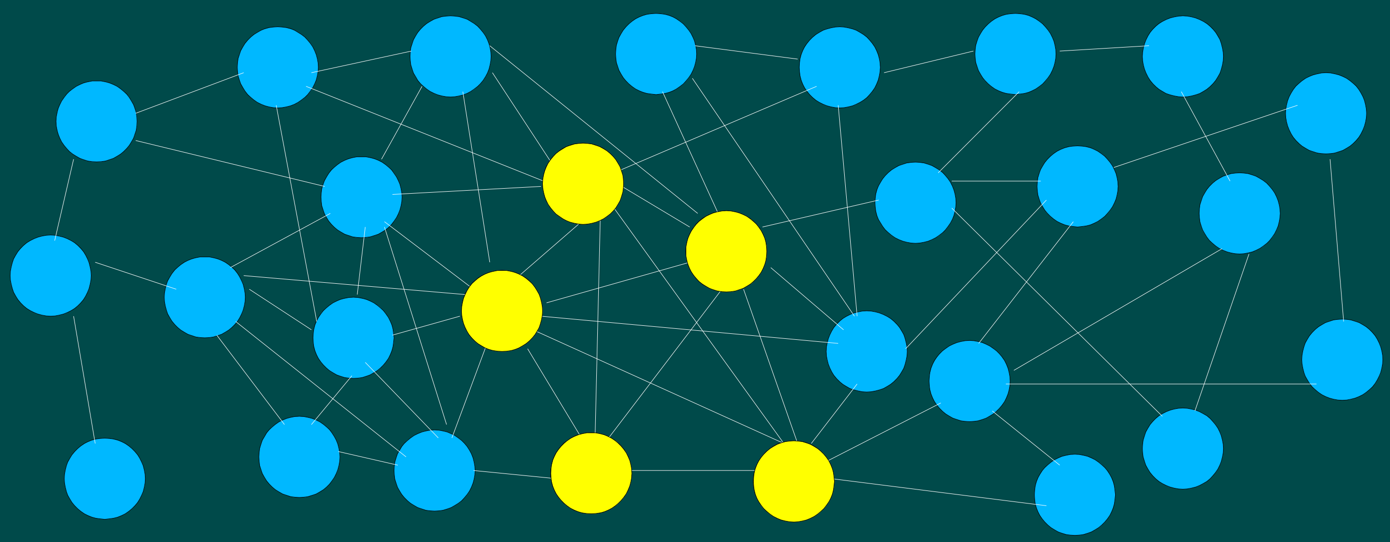
- 最大クリーク問題解説
  - 研究の目的と背景
  - 用語と定義
- 解を求めるアルゴリズム解説
  - 伝統的な基本アルゴリズム
  - 電気通信大学富田教授グループのアルゴリズム
  - 独自に考えたアルゴリズム

- 最大クリーク問題解説
  - 研究の目的と背景
  - 用語と定義

- 解を求めるアルゴリズム解説
  - 伝統的な基本アルゴリズム
  - 電気通信大学富田教授グループのアルゴリズム
  - 独自に考えたアルゴリズム

最大クリーク問題とは

- 無向グラフ中の部分グラフの中で、完全グラフでサイズが最大のものを選び出す、組み合わせ最適化問題の一種
- NP 困難…グラフサイズが大きい→計算時間が膨大



↑ 黄色が最大クリーク

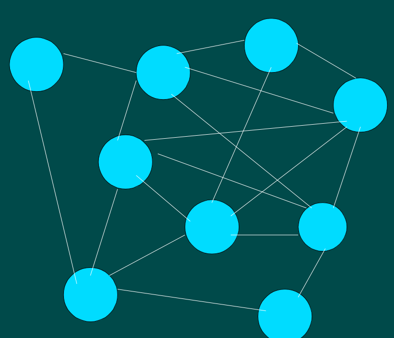
研究の背景と目的

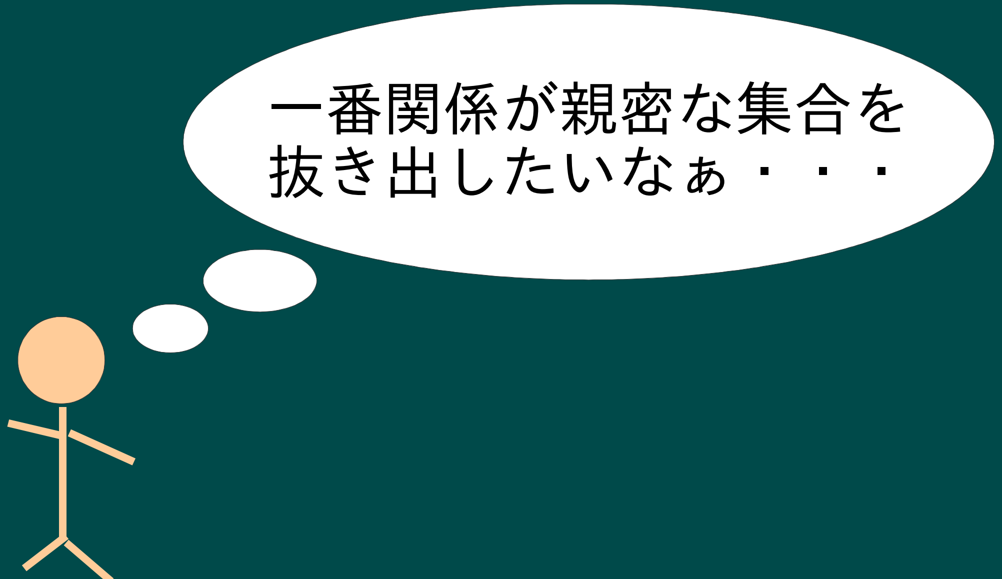
- NP困難…高速なアルゴリズムの存在が重要
- グラフ理論は応用の幅が広い
- 大きいグラフ…計算不能

↓

計算可能にする

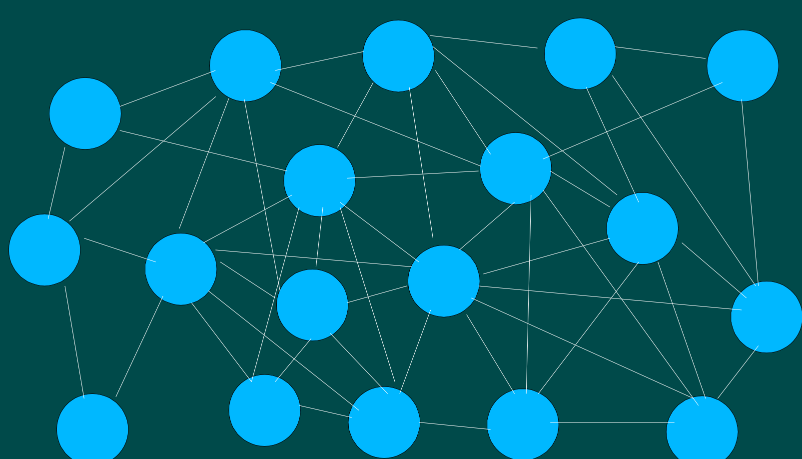
何かのデータの関係を
グラフとして表した



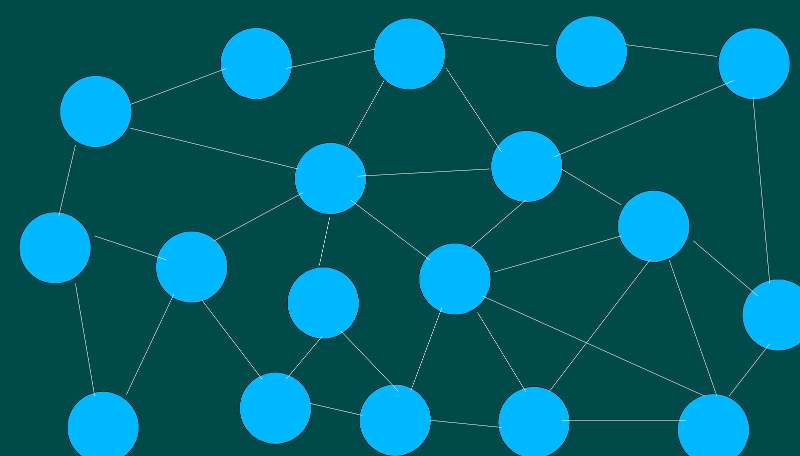


研究の方針

- 従来のアルゴリズムを解析
  - 長所・短所・改良点？
- 従来アルゴリズム VS 改良アルゴリズム
  - 処理時間比較
- 各アルゴリズムがどのようなグラフに対して有効か



密なグラフで速いか？

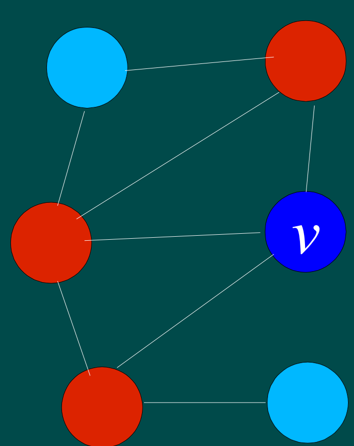


疎なグラフで速いか？

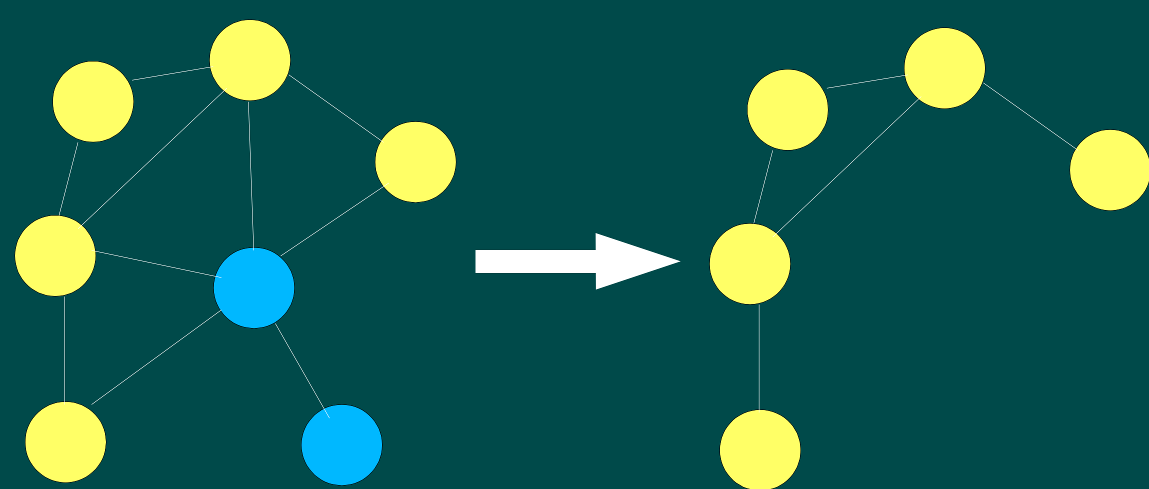
- 最大クリーク問題解説
  - 研究の目的と背景
  - 用語と定義
- 解を求めるアルゴリズム解説
  - 伝統的な基本アルゴリズム
  - 電気通信大学富田教授グループのアルゴリズム
  - 独自に考えたアルゴリズム

用語

- 近傍：ある頂点に隣接する頂点全ての集合
- 点誘導部分グラフ：あるグラフから適当な頂点集合を選んで、その頂点集合とそれらの間に存在する辺で構成されるグラフ



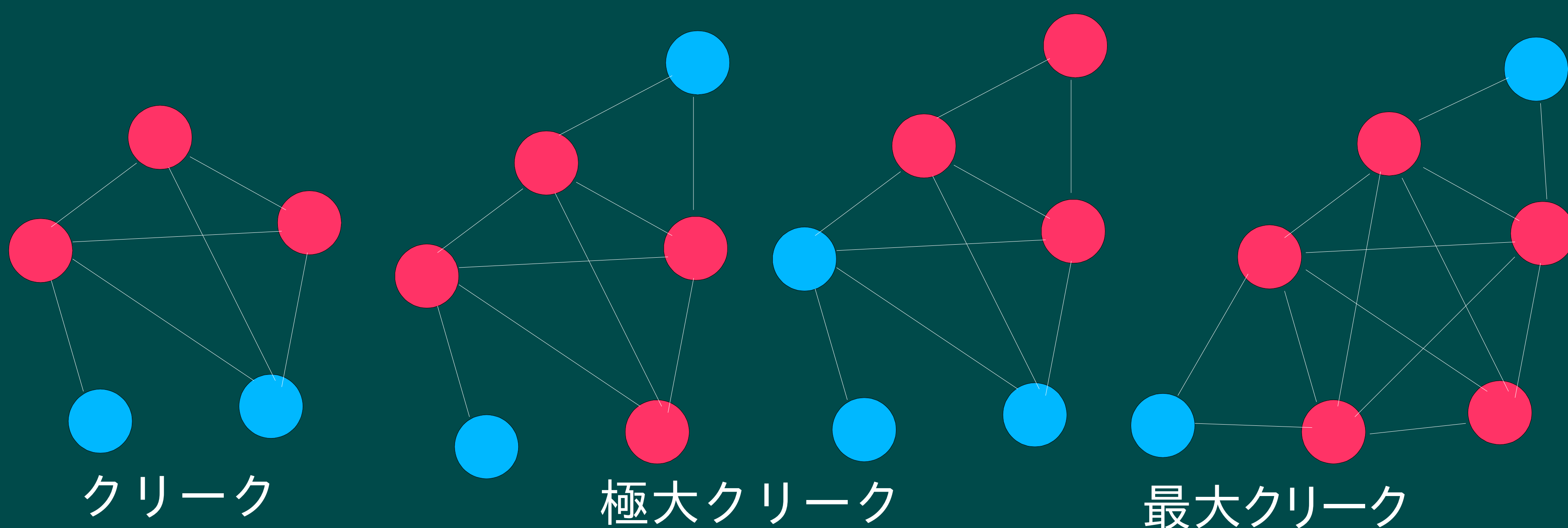
：v の近傍



による点誘導部分グラフ

用語

- クリーク：完全グラフを誘導する頂点集合
- 極大クリーク：他のクリークの真部分集合でないクリーク
- 最大クリーク：グラフ中で頂点数が最大のクリーク

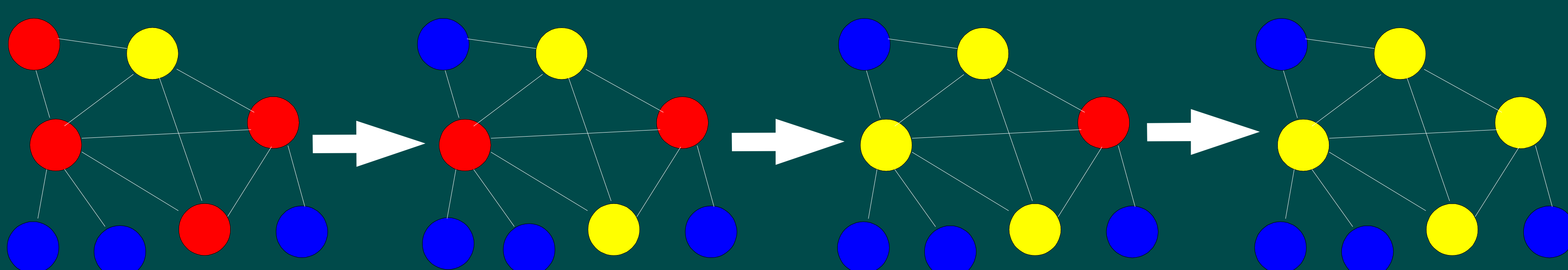


- 最大クリーク問題解説
  - 研究の目的と背景
  - 用語と定義
- 解を求めるアルゴリズム解説
  - 伝統的な基本アルゴリズム
  - 電気通信大学富田教授グループのアルゴリズム
  - 独自に考えたアルゴリズム

基本アルゴリズム

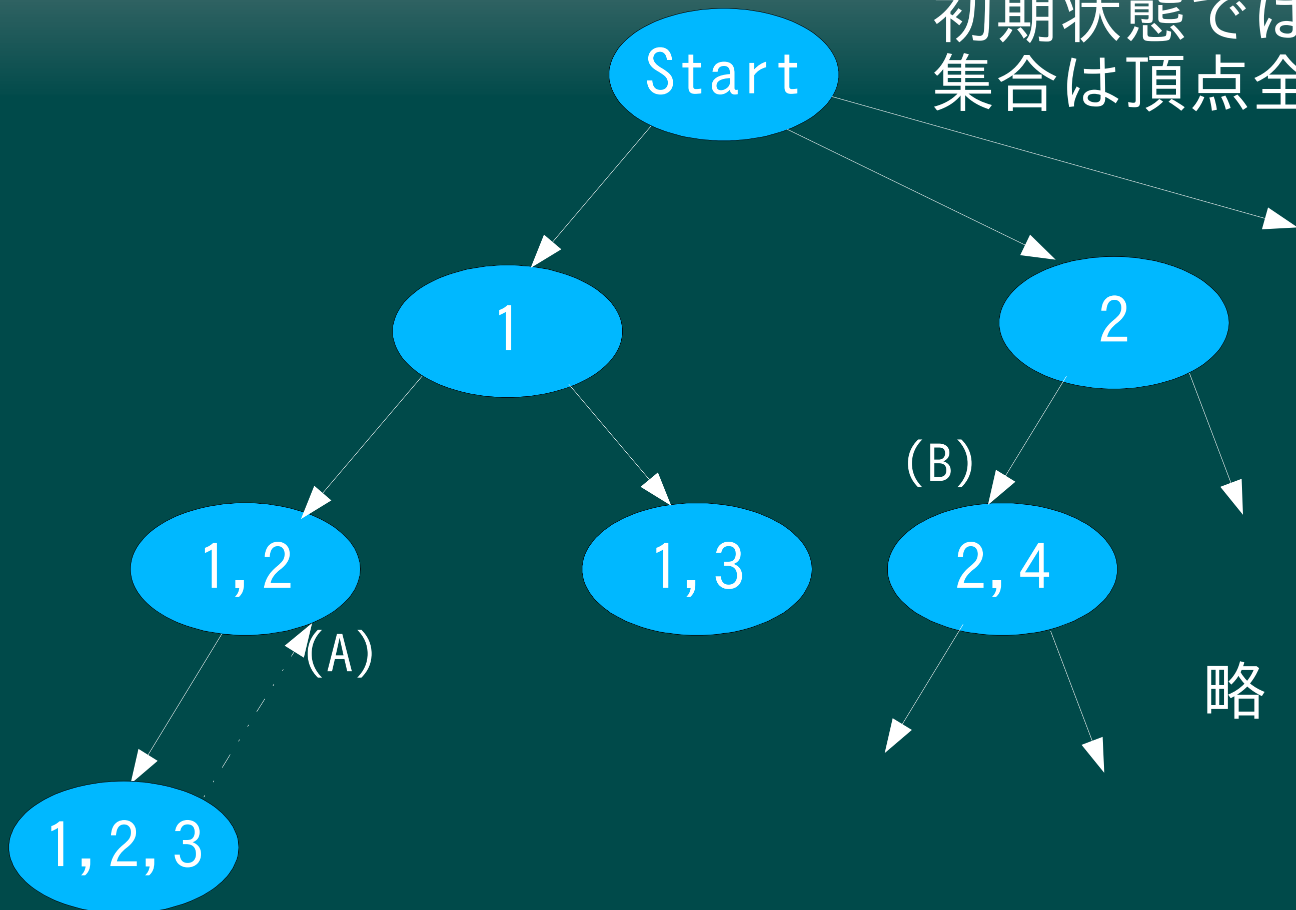
- 1990 年 R.Carraghan と P.Paradalos によって提唱された単純なアルゴリズム
- 「候補節点集合」：ある時点で保持しているクリークに付け加えても、またクリークとなるような頂点の集合
- 候補接点集合中の頂点をクリークに追加→サイズ+1 のクリークを作って候補節点集合を更新

●：候補節点集合　●：保持するクリーク

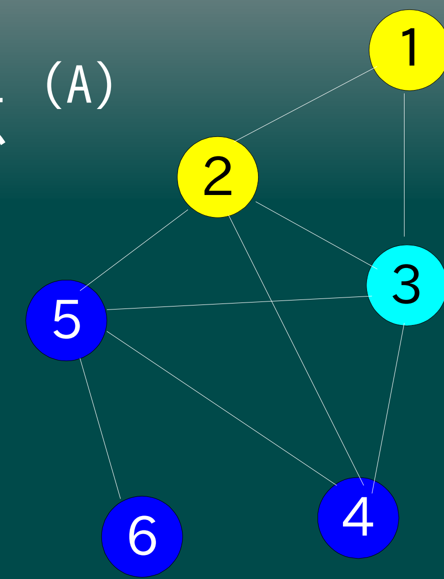


探索木

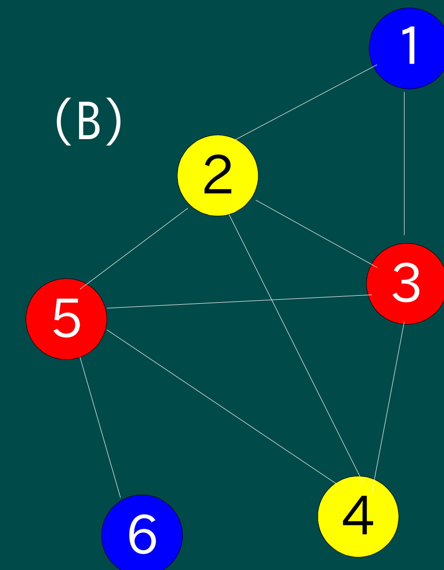
初期状態では候補接点集合は頂点全体



(A)



(B)



探索木全体を書くと、頂点数の割りに膨大なサイズになる

: 候補接点集合　: 保持するクリーク　: 候補接点集合から取り除いたノード

- 最大クリーク問題解説
  - 研究の目的と背景
  - 用語と定義

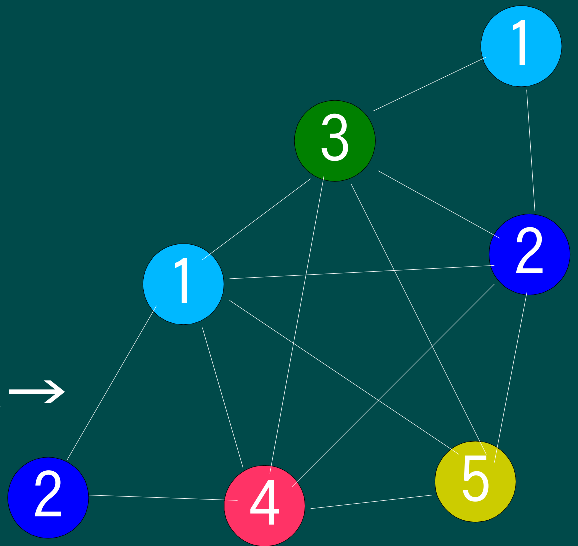
- 解を求めるアルゴリズム解説
  - 伝統的な基本アルゴリズム
  - 電気通信大学富田教授グループのアルゴリズム
  - 独自に考えたアルゴリズム

彩色とは

- 各頂点に番号を割り振り、隣接する頂点には同じ番号を割り振らないようにする

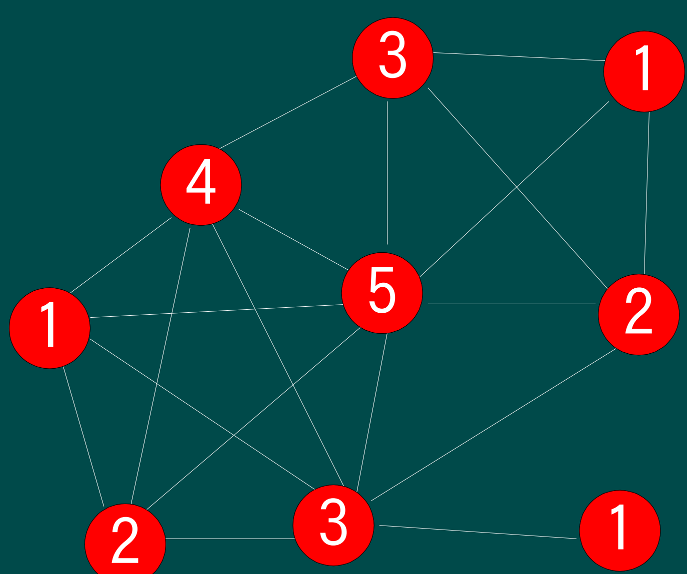
- k 彩色問題… NP 困難
- 近似彩色…多項式時間 (3 近似)

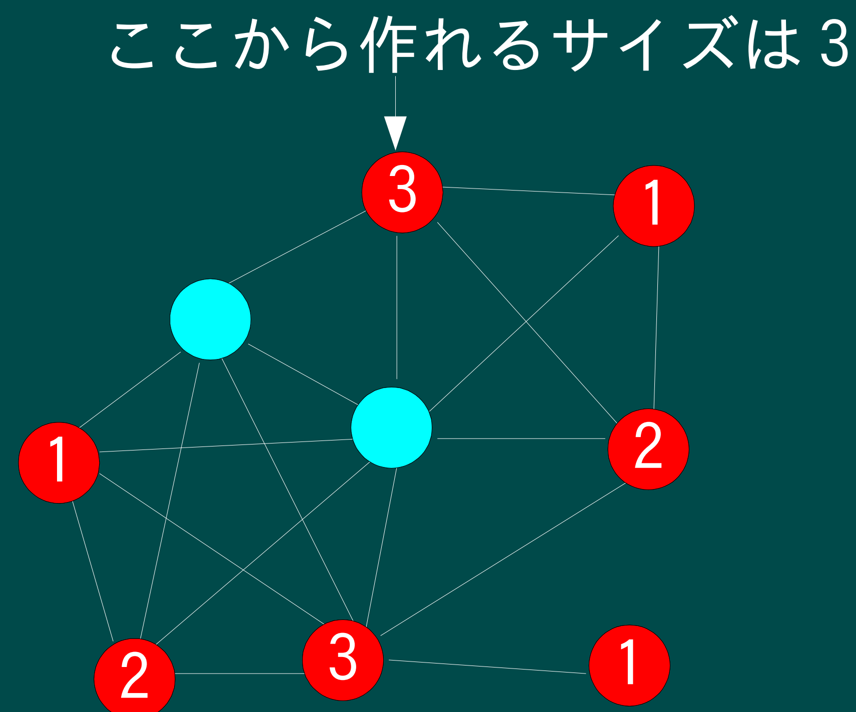
5 色で彩色→

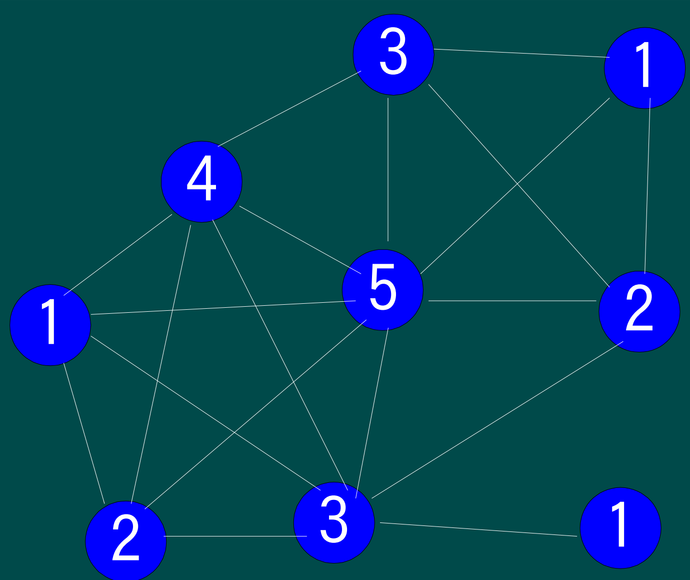


彩色による分岐限定 （富田教授のアルゴリズム）

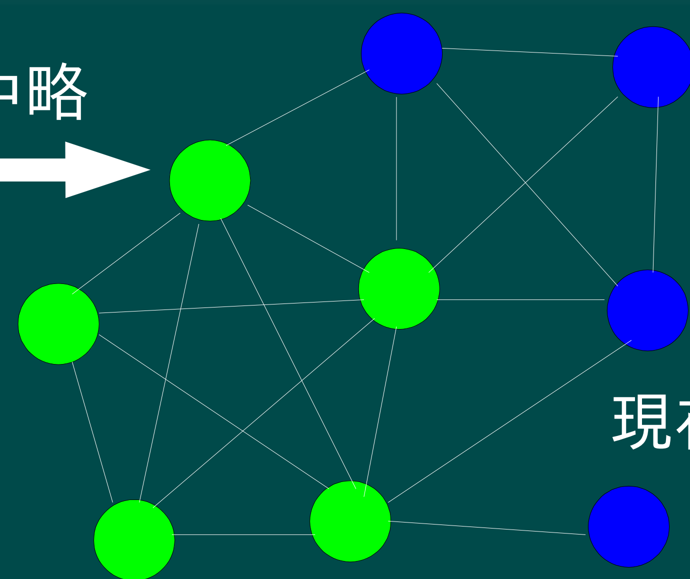
- 割り振られた番号の大きい順に、候補接点集合から取り出す
- 割り振られた番号は、その頂点を中心として候補接点集合内でできるクリークサイズの上界





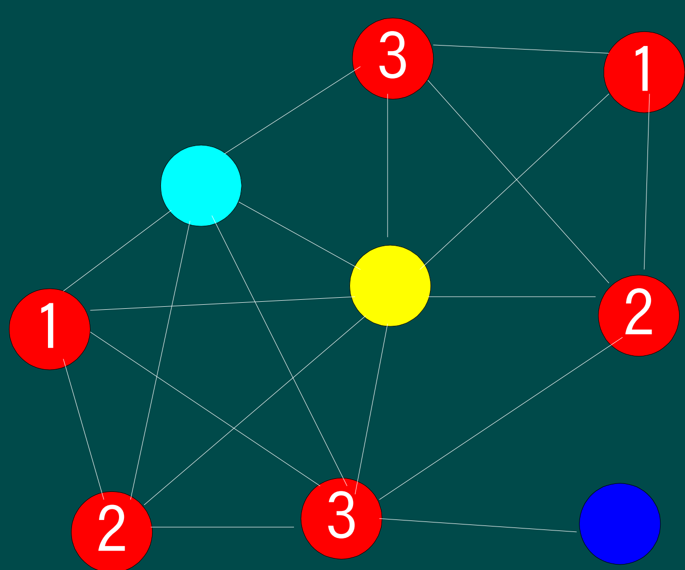


中略



保持する現在最大のクリーク

現在の最大サイズ :5



1 にサイズ 3 を加えても、
現在最大の 5 には届かない

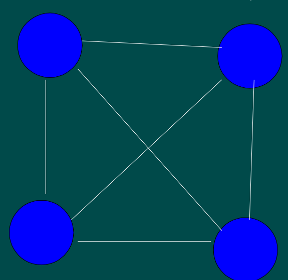
探索木深さ 1 の時点で 6 個の子を探索打ち切りできた

これらの頂点で作る極大クリークが全列挙される過程を全て飛ばす

- 最大クリーク問題解説
  - 研究の目的と背景
  - 用語と定義
- 解を求めるアルゴリズム解説
  - 伝統的な基本アルゴリズム
  - 電気通信大学富田教授グループのアルゴリズム
  - 独自に考えたアルゴリズム

独自開発のアルゴリズム（上界を下げる）

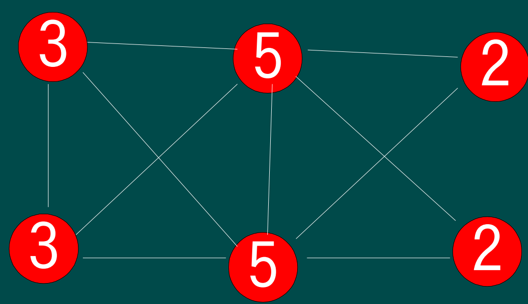
- サイズN以上のクリークを作るには、次数$N-1$以上のノードがN個以上必要



サイズ4→ 各次数3以上

- 候補節点集合を次数について降順で整列し、i が i 番目のノードの次数+1以上となる最小の変数 i に対して、i 番目のノードの次数+1は候補節点集合中のクリークサイズの上界となる

<u>非常に短時間で頂点集合のクリークサイズの上界を求めることができる</u>



候補接点集合を次数に付いて降順で整列

index	1	2	3	4	5	6
degree	5	5	3	3	2	2

↑ 上界 [illegible] 4 個

この候補節点集合からできるクリークサイズの上界は 4

独自開発のアルゴリズム（隣接ノード上界）

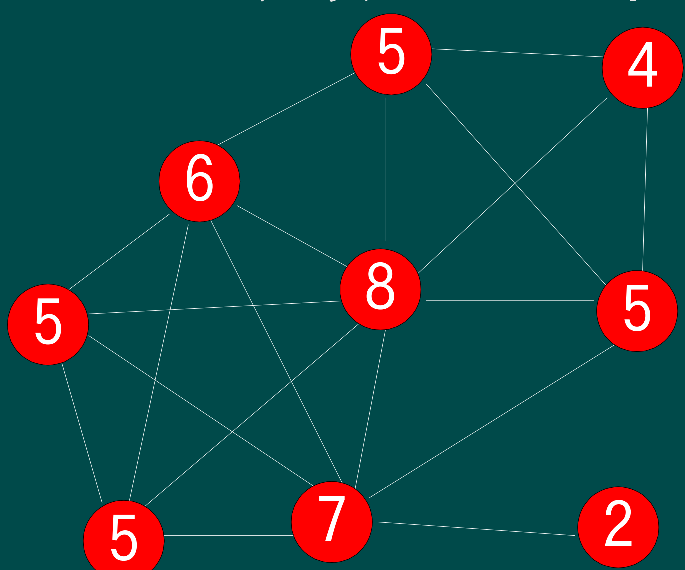
- 各頂点 v に変数 $\mathrm{addeg}(v)$ を割り当てる

- v の近傍と v の和集合を次数について降順で整列し、i が i 番目のノードの次数+1以上となる最小の変数 i に対して、i 番目のノードの次数+1を $\mathrm{addeg}(v)$ とする

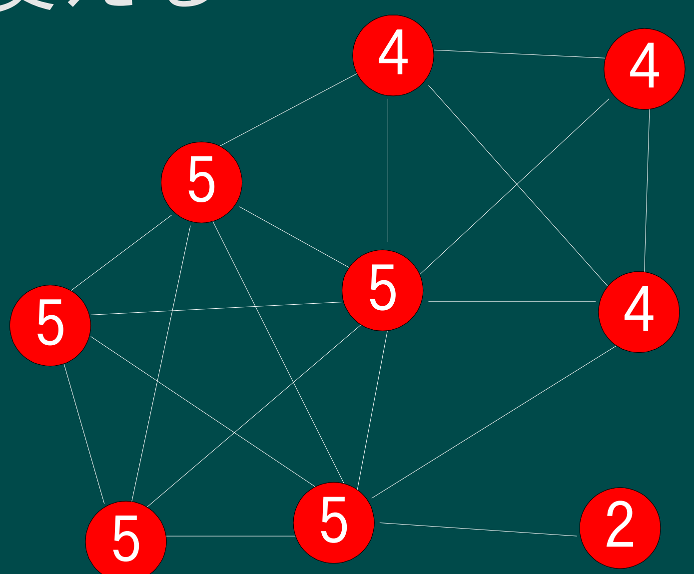
- $\mathrm{addeg}(v)$ は v を含んだクリークサイズの上界となる

独自開発のアルゴリズム（隣接ノード上界）

- addeg の大きい順に候補節点集合から取り出して、保持するクリークに加える
- 取り出す頂点の addeg に基づいて探索打ち切りが可能
- 先程の候補節点集合の上界を求める際に、次数 +1 の代わりに使える



次数 +1 で見る各ノードを含むクリークサイズの上界



addeg で見る各ノードを含むクリークサイズの上界

比較結果と考察

- 対象データ :DIMACS ベンチマークグラフ
  - 頂点数 50-1500 、辺密度 0.05-0.99 程度
- 実行環境
  - OS: Linux version 2.6.9-1.6_FC2
  - CPU: Intel Pentium4 2.66GHz
  - RAM: 1024MB
  - コンパイラ : g++ 3.3.3
  - C++ クラスライブラリ LEDA4.5 を使用して実装

（富田教授のアルゴリズムは自分で実装したもので、
実際の富田教授のグループの実験結果よりも遅いです）

比較結果と考察

	頂点数	辺数	辺密度	$\|Q_{max}\|$	富田	原田	基本
×	45	918	0.93	16	0.92	20min	14min
×	171	9435	0.64	11	39.36	4min	7min
○	200	3235	0.16	24	0.11	0.07	7min
○	300	10933	0.24	8	2.4	0.94	5.3
○	500	9139	0.07	26	0.4	0.25	Over1h
○	200	8473	0.43	58	1.06	0.19	?
○	1000	122253	0.24	10	26min	4min	?
○	500	23191	0.19	64	3.72	0.48	?
○	500	46627	0.37	126	26.29	1.6	?
○	1500	284923	0.25	11	Over2h	49min	?

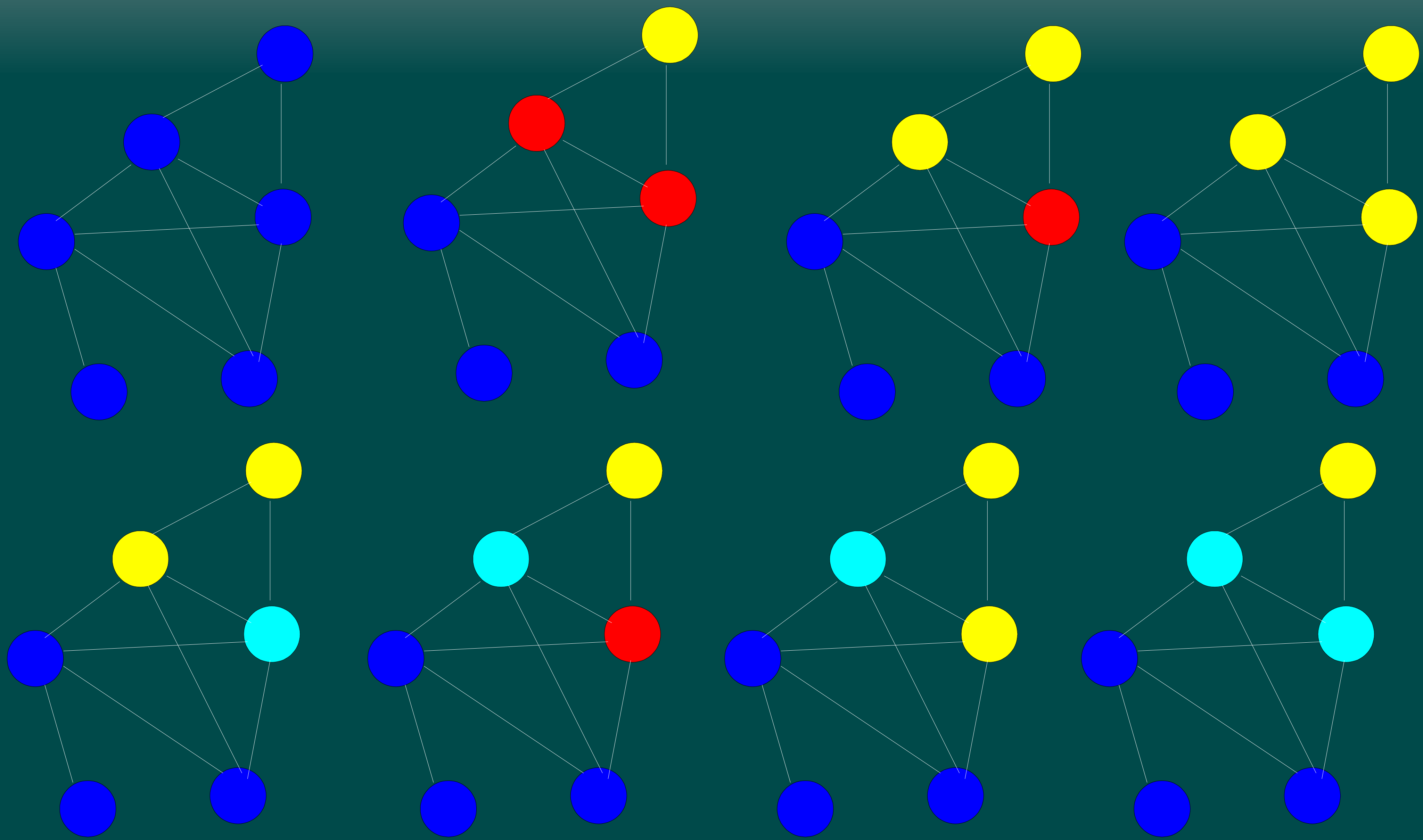
疎なグラフにそこそこ強く、密なグラフにとんでもなく弱い
（密なグラフに強いアルゴリズムは多いらしい）

良い所の主張

- 各ノードが「そのノードを含むクリークサイズの上界」のデータを持っていれば、応用可能である。
- 候補節点集合に対する上界の探索にかかる手間が非常に少なく、２回目以降の探索は続きからできるので、手間の割に効果のあるアルゴリズムができた。
- 調べた限りでは、全く新しい自分独自の実用的なアルゴリズムである。

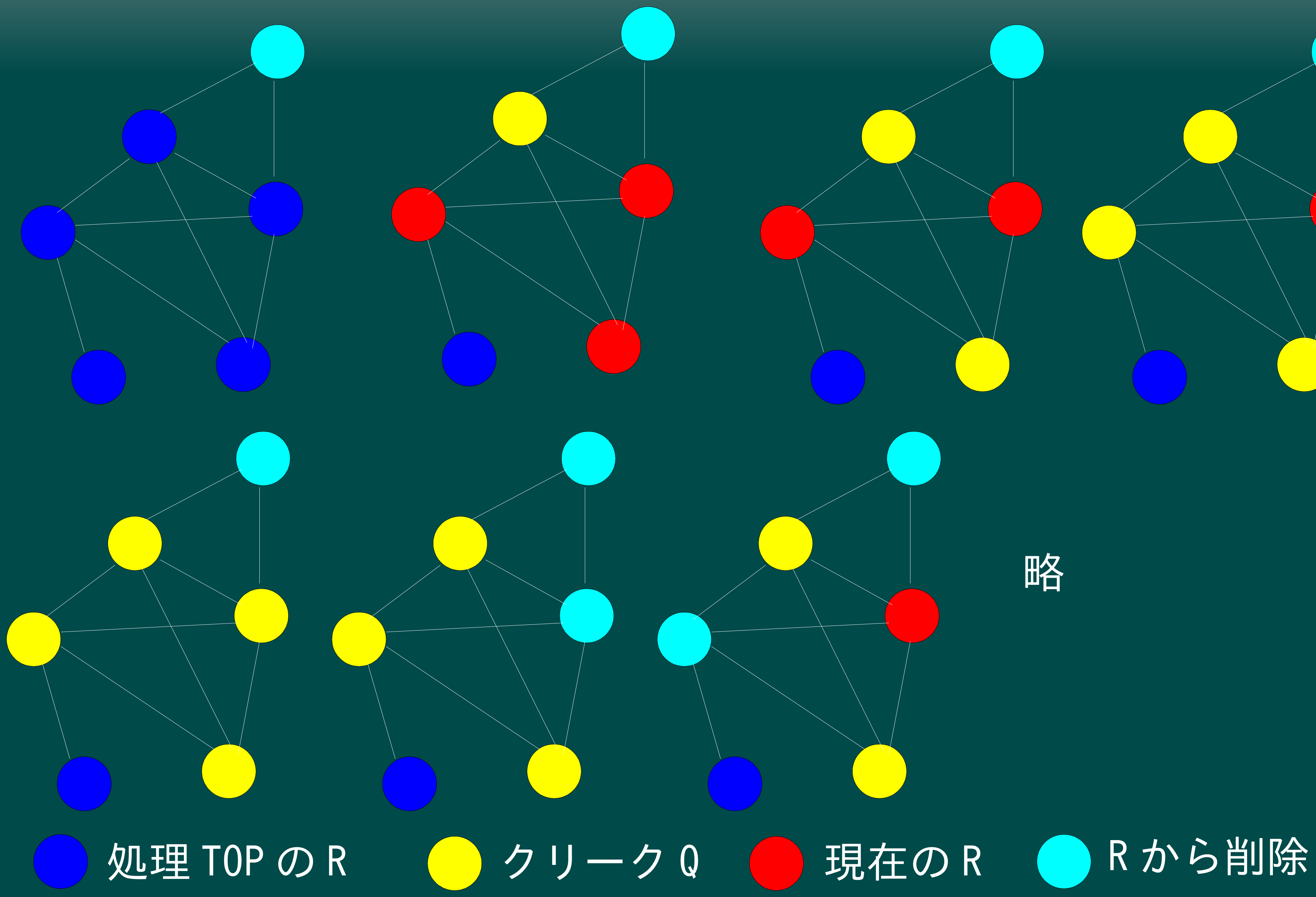
終わり

動作の図解



処理 TOP の R　クリーク Q　現在の R　R から削除

動作の図解



次数による分岐限定（江口氏のアルゴリズム）

- 次数 $N-1$ 未満の頂点を含むクリークは、サイズ N 以上のクリークとなることはない

  ↓
- $|Q_{max}|$ 未満の次数の頂点を削除していくことにより、グラフ自体を小さくしていく
- $\mathrm{degree}(p) \leq |Q_{max}|$ となる頂点 p を候補接点集合から選ばない。

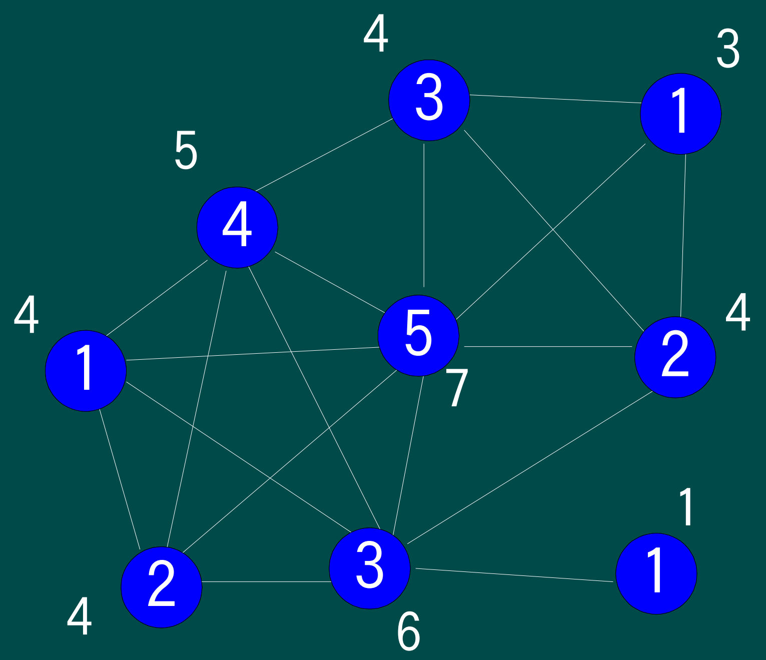
- 富田教授のアルゴリズムに上記の 2 つの処理を加えて高速化を図る。

- 最大クリーク問題解説
  - 研究の目的と背景
  - 用語と定義

- 解を求めるアルゴリズム解説
  - 伝統的な基本アルゴリズム
  - 電気通信大学富田教授のアルゴリズム
  - 中央大学江口氏のアルゴリズム
  - 独自に考えたアルゴリズム

彩色による分岐限定（富田教授のアルゴリズム）

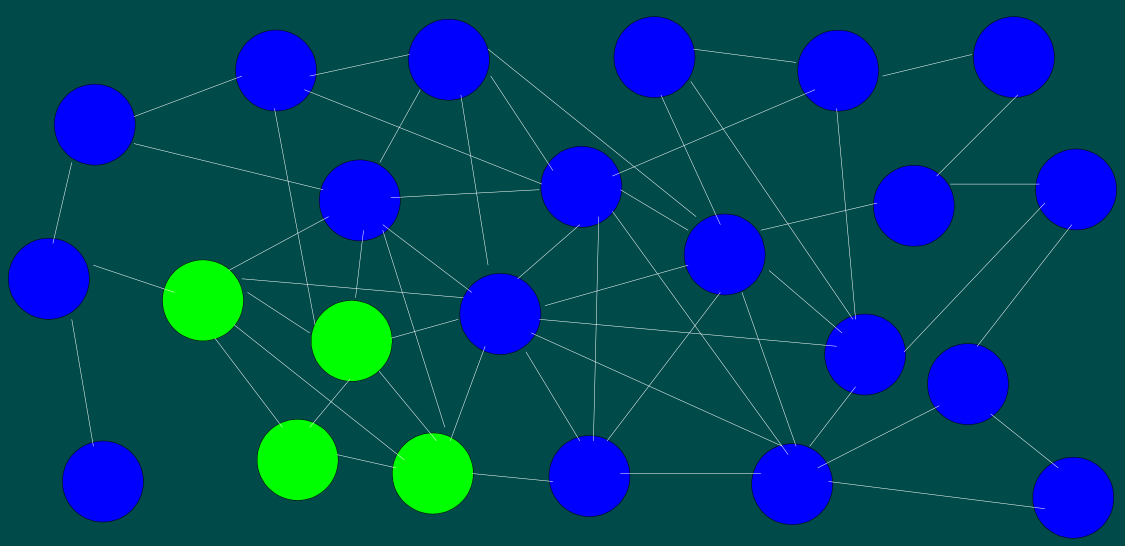
1. 次数の小さい頂点から順に、隣接する頂点同士には同じ番号を割り振らないように、できるだけ小さい番号を割り振る
2. 近似彩色になる（確実には最小にならず）



- 大きいクリークを早いうちに見つける

↓

- 候補接点集合を取り出す順番を変える
  - グラフに基準となる情報の付加
    （使用メモリ増大・処理高速化）



サイズ４を既に見つけた

↓

上界４以下の頂点と頂点
集合を全て無視できる