情報科学入門
「コンピュータにどう考えさせるか？」

九州大学 システム情報科学研究院
准教授　稲永 俊介

九州大学
KYUSHU UNIVERSITY

KYUSHU UNIVERSITY 100th 2011
知の新世紀を拓く

もくじ

1. 自己紹介
2. 高校時代
3. 情報科学とは？
4. 研究紹介
5. まとめ

自己紹介

自己紹介

稲永 俊介（いねなが しゅんすけ）

出身：福岡県糟屋郡志免町

学歴：福岡高校 ⇒ 九州工業大学 ⇒ 九州大学大学院

職歴：科学技術振興機構 ⇒ ヘルシンキ大学 ⇒
京都大学 ⇒ 日本学術振興会 ⇒ 九州大学

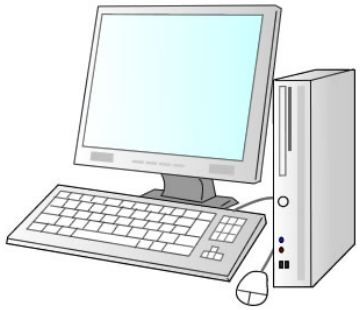
自己紹介（つづき）

現職：九州大学 システム情報科学研究院 准教授
専門：情報科学

平日：学生と共に日々研究に励む
　　授業もする（理学部，工学部）

情報科学とは？

情報科学って何？



- コンピュータ上で行われていることは
  すべて「計算」である.
- その「計算」と上手に付き合ったり、
  「計算」をうまく制御するには、
  「計算」とは何であるかを理解する必要がある.
- そのための科学が
  Computer Science（**情報科学**）
- コンピュータ上で何らかの「計算」を行うとき、
  アルゴリズムの設計が重要となる.

アルゴリズムって？

- コンピュータは、四則演算（＋，－，×，÷）などの単純な計算を高速かつ正確に行える
- しかし、複雑な問題を解くための「手順」をコンピュータ自身が編み出すことはできない
- 単純な計算を組み合わせて、より複雑な問題を解くための手順のことを**アルゴリズム**という
- 高速なアルゴリズムの開発は人間（研究者）の役割！

最大公約数問題

問題

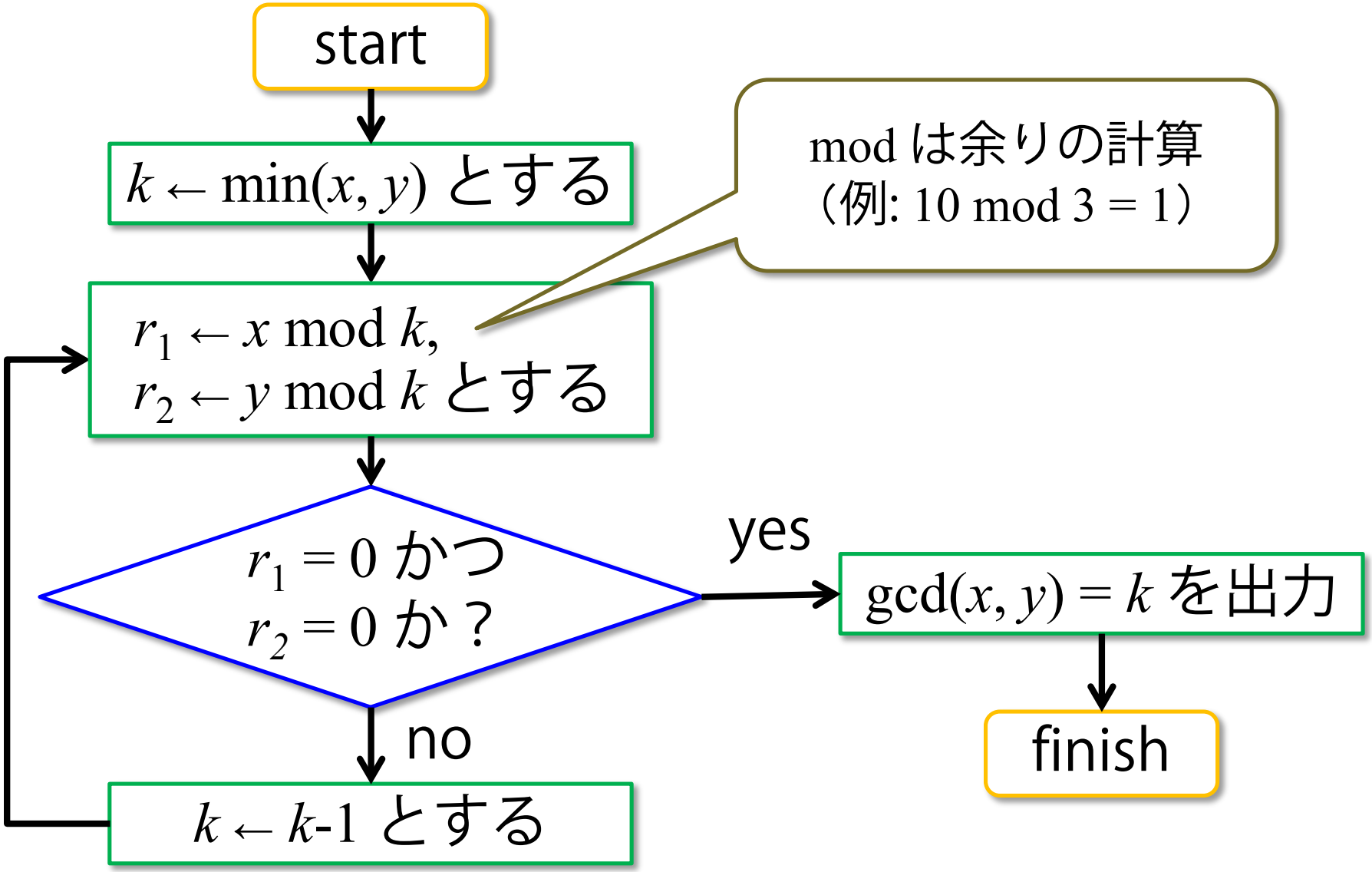
正整数 x と y が与えられたとき、
x と y の最大公約数 $\gcd(x, y)$ を求めよ.

gcd : greatest common divisor

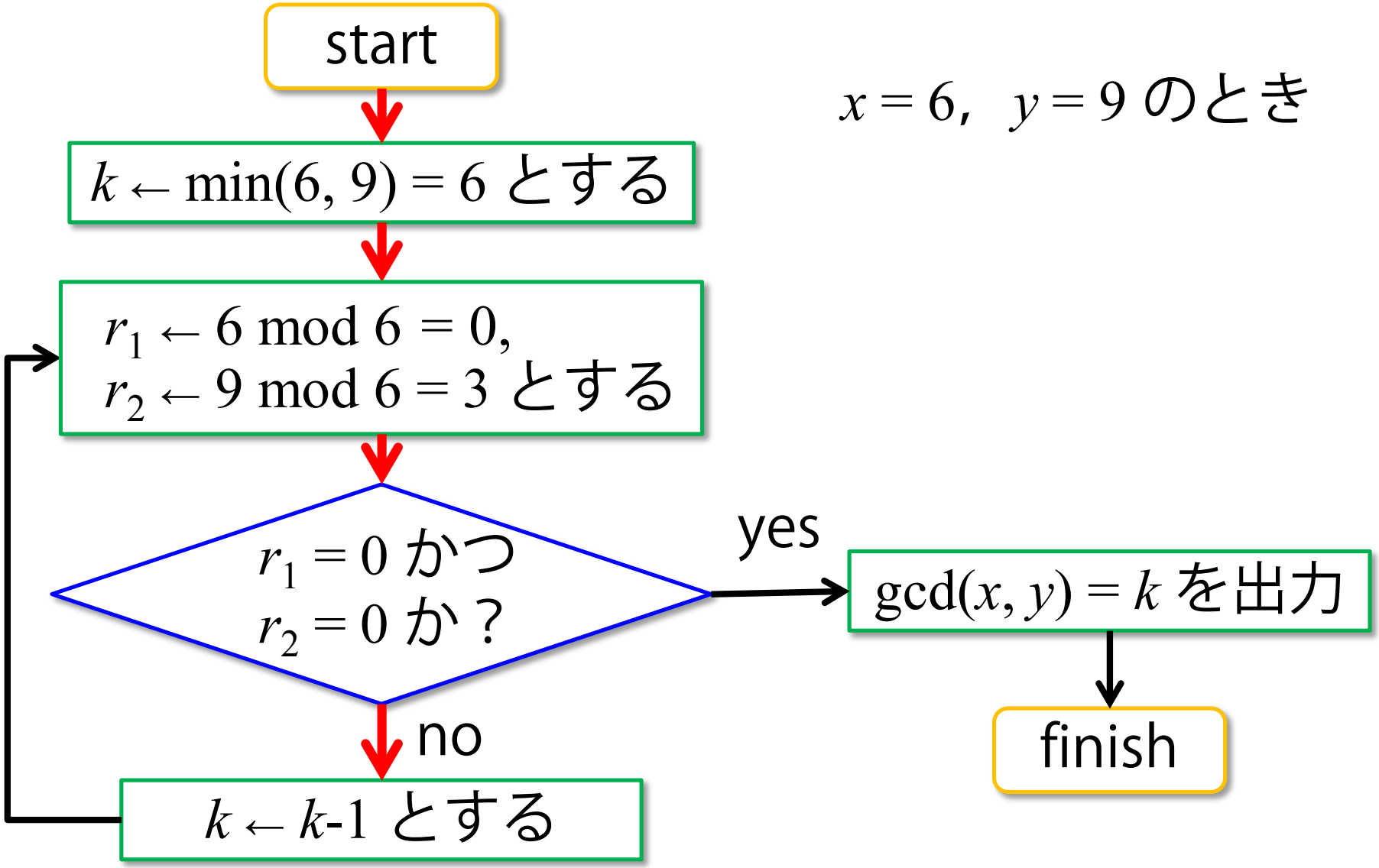
例

- $\gcd(16, 12) = 4$
- $\gcd(63, 14) = 7$
- $\gcd(47, 32) = 1$

しらみつぶし的アルゴリズム naive_gcd

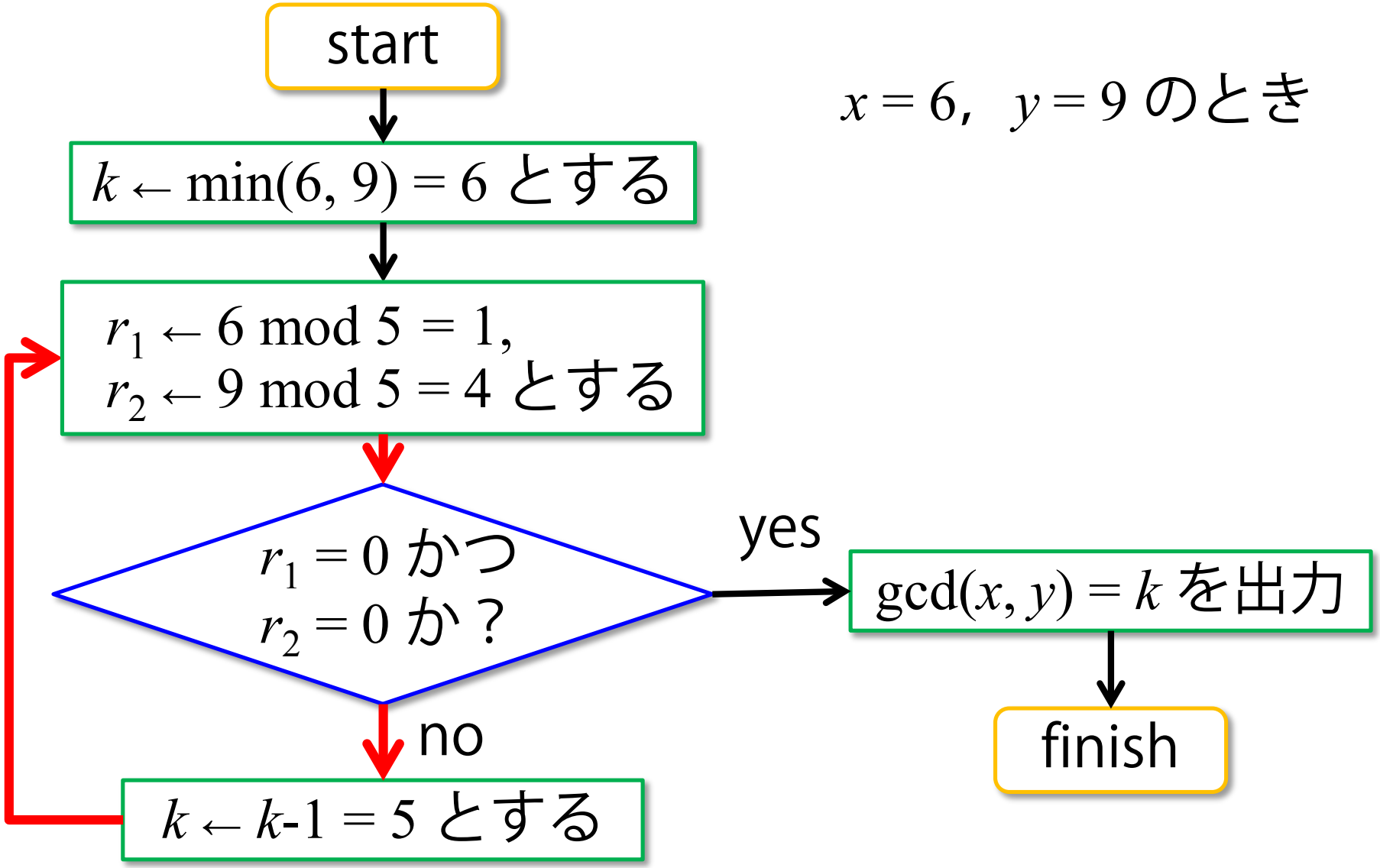


naive_gcd の実行例



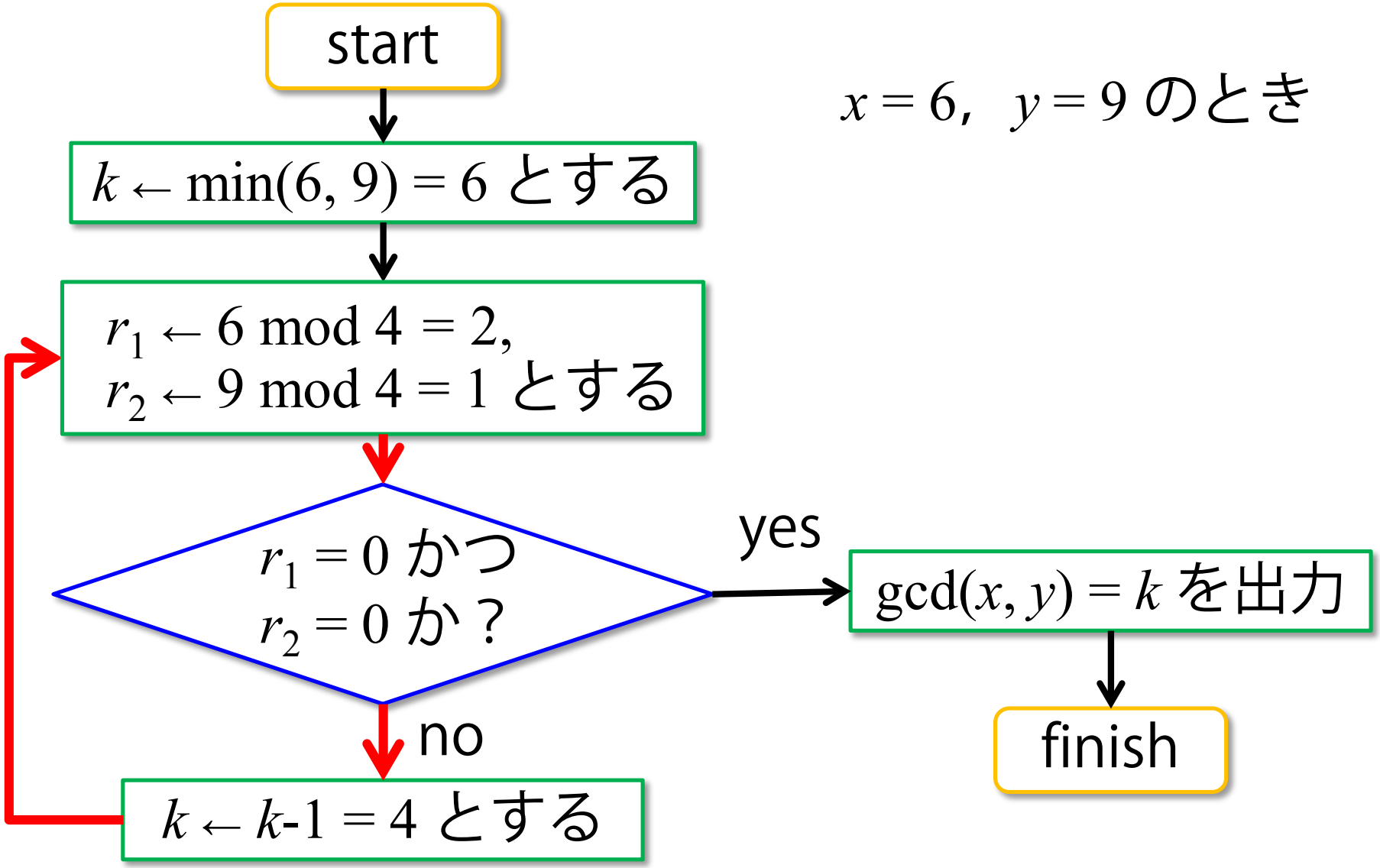
$x = 6,\ \ y = 9$ のとき

naive_gcd の実行例



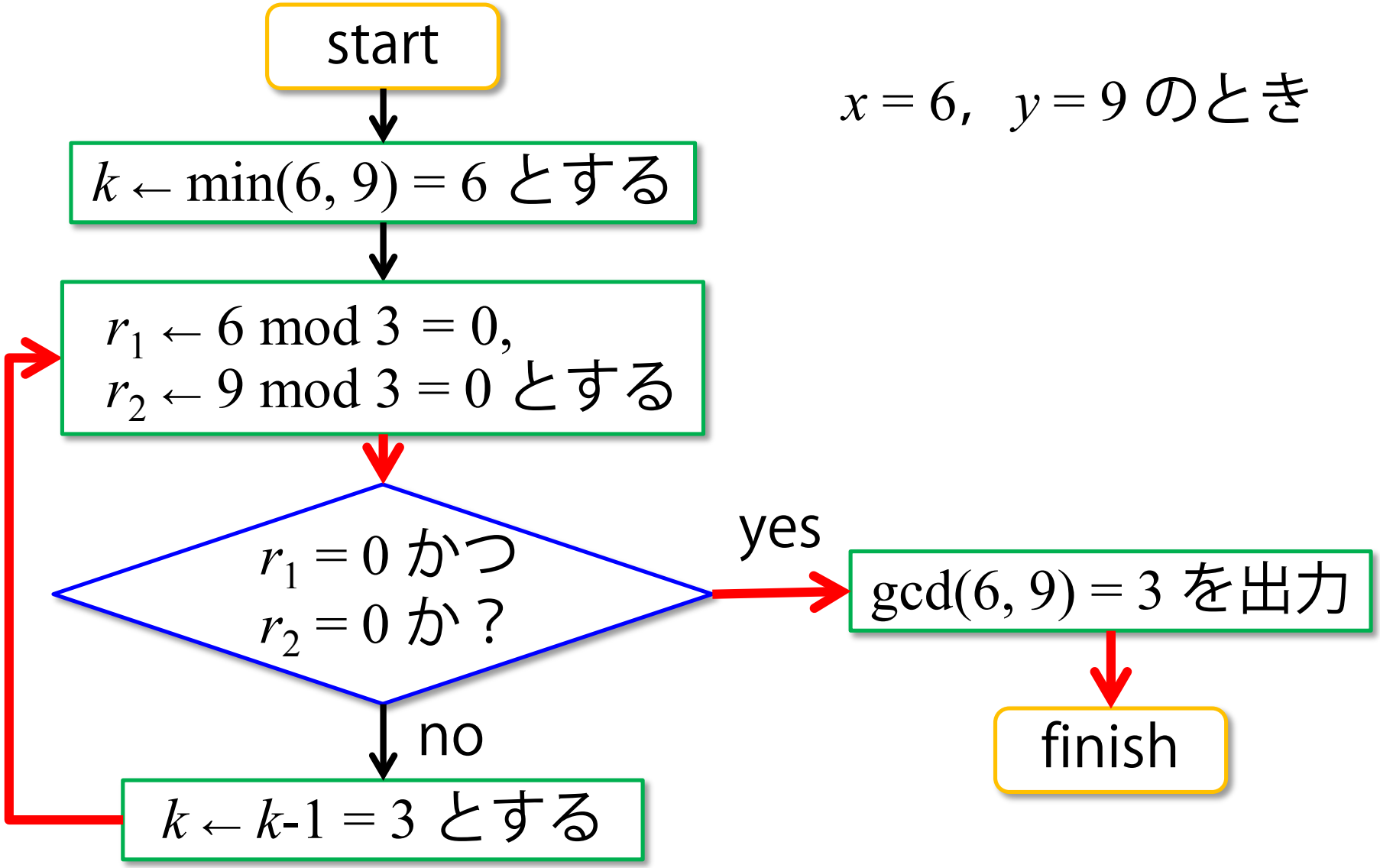
$x = 6,\ \ y = 9$ のとき

naive_gcd の実行例



$x = 6,\ \ y = 9$ のとき

naive_gcd の実行例



$x = 6,\ \ y = 9$ のとき

naive_gcd の計算時間見積もり

- $n = \min(x, y)$ とすると，余り (mod) の計算回数は最悪の場合で $2n$ 回
- 余り (mod) の計算の1回の計算時間
  - PC 1ステップ 10^{-8}秒
  - スーパーコンピュータ 1ステップ 10^{-11}秒

n	パソコン	スパコン
10000	0.2×10^{-3} 秒	0.2×10^{-6} 秒
100万	0.02 秒	0.2×10^{-4} 秒
1億	2 秒	0.002 秒
1兆	5.55 時間	20 秒
1京	6.3 年	55.5 時間
10^{50}	莫大	6.3×10^{31} 年
10^{100} ($\approx 2^{332}$)	莫大	6.3×10^{81} 年

高速化への道

最大公約数問題を，
「長方形をなるべく大きな正方形で埋める問題」
に置き換えて考える．

問題

縦の長さ x，横の長さ y の長方形が与えられたとき，長方形を敷き詰める最大の正方形を求めよ．

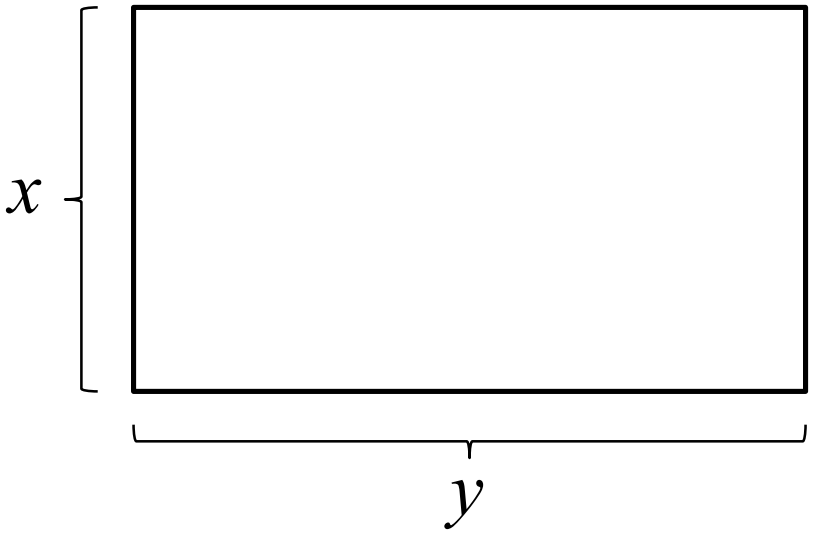
x, y は正整数

長方形敷き詰め問題

問題

縦の長さ x，横の長さ y の長方形が与えられたとき，長方形を敷き詰める最大の正方形を求めよ．

x, y は正整数

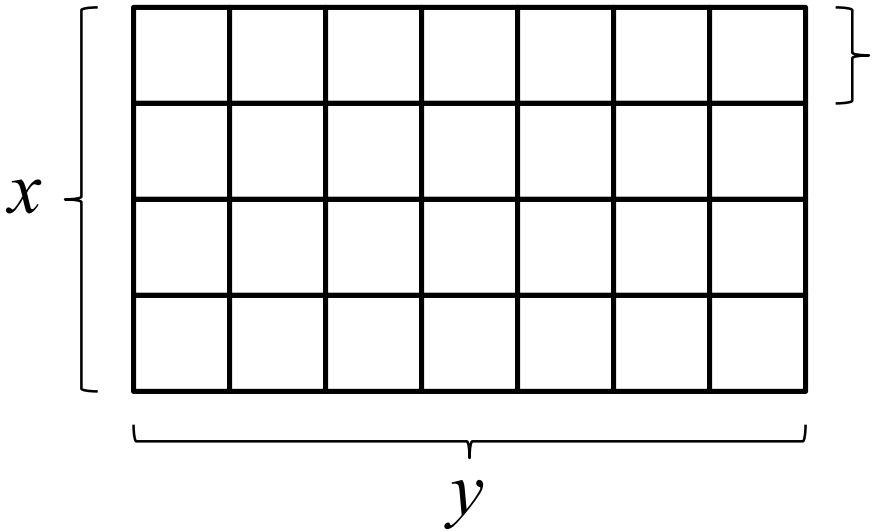


長方形敷き詰め問題

問題

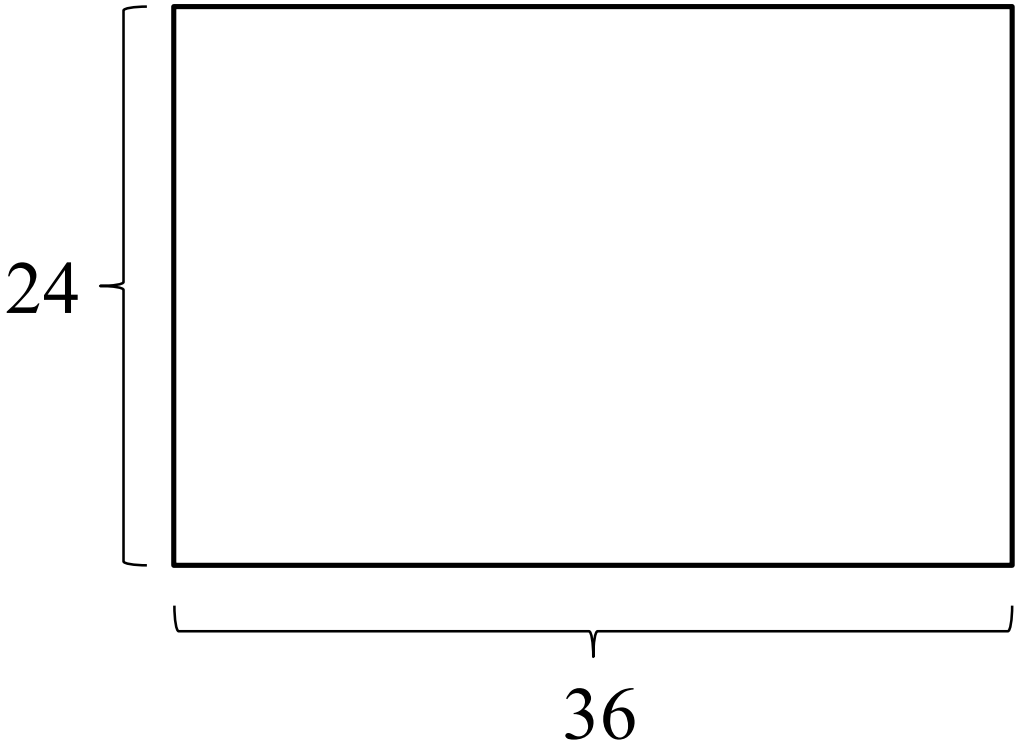
縦の長さ x，横の長さ y の長方形が与えられたとき，長方形を敷き詰める最大の正方形を求めよ．

x, y は正整数



- 長方形を埋める正方形の一辺の長さ＝ x と y の公約数
- そのような最大の正方形の一辺の長さ = $\gcd(x, y)$

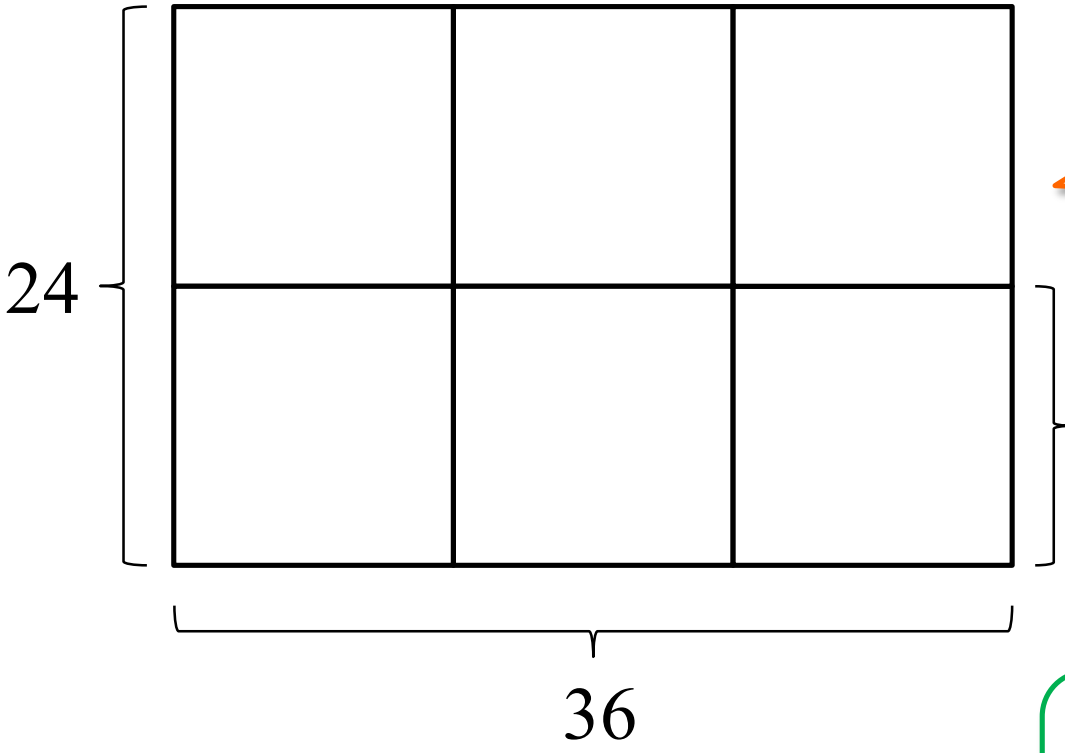
長方形敷き詰め問題(例)



24 と 36 の最大公約数
gcd(24, 36) を求めたい

- 長方形を埋める正方形の
一辺の長さ＝ x と y の公約数
- そのような最大の正方形の
一辺の長さ = $\gcd(x, y)$

長方形敷き詰め問題(例)

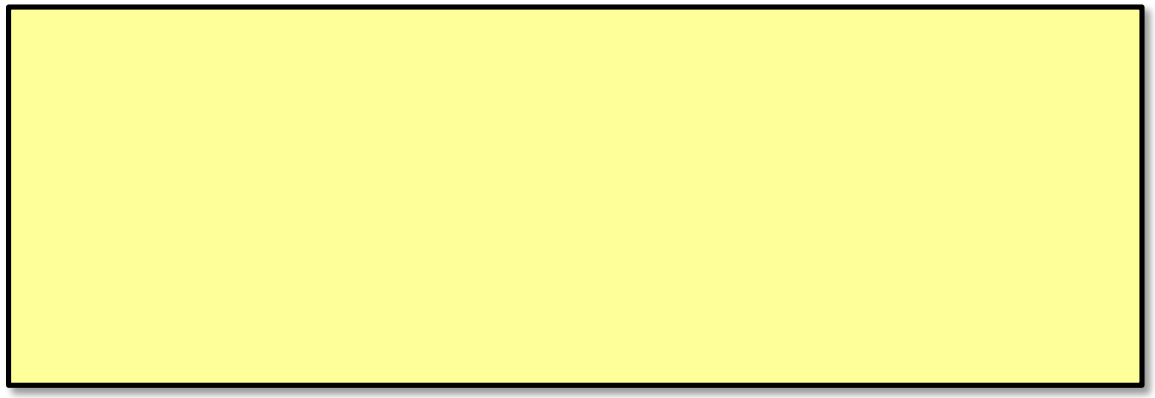


24 と 36 の最大公約数
gcd(24, 36) を求めたい

- 長方形を埋める正方形の
一辺の長さ＝ x と y の公約数
- そのような最大の正方形の
一辺の長さ = gcd(x, y)

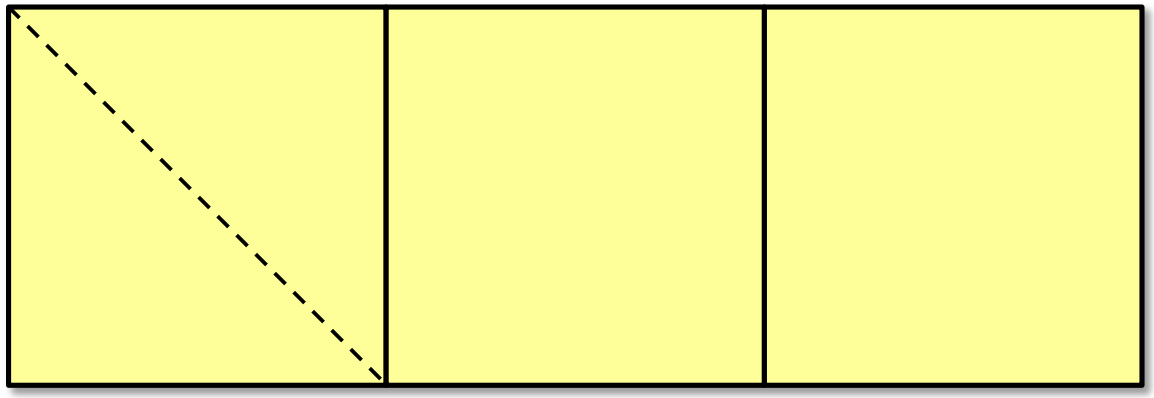
演習 1

- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



解答

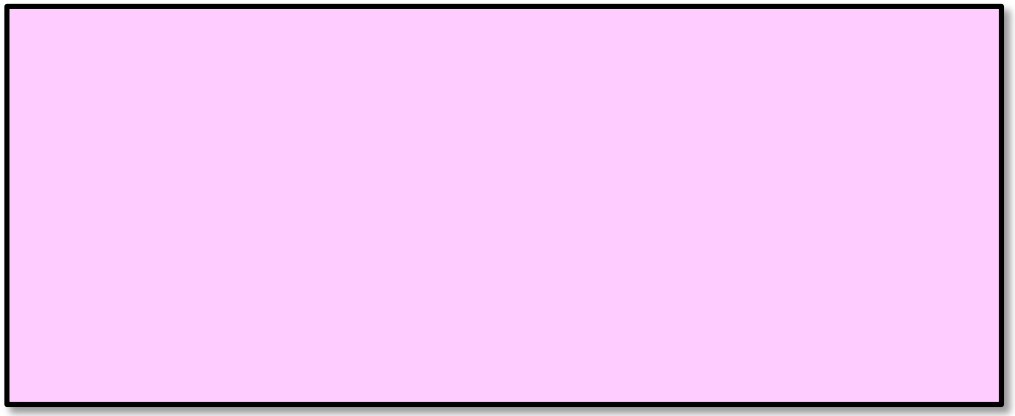
- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



長い辺を短い辺で割る.
ぴったり割れたら，短い辺の長さが最大公約数.

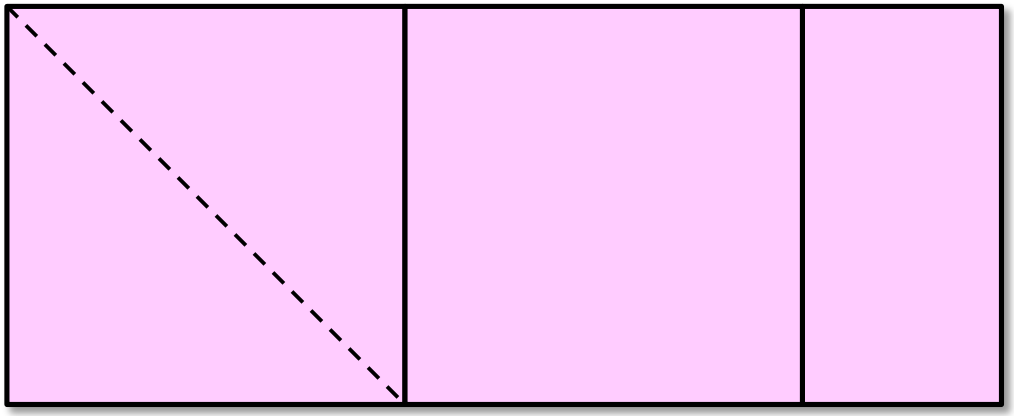
演習2

- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



解答

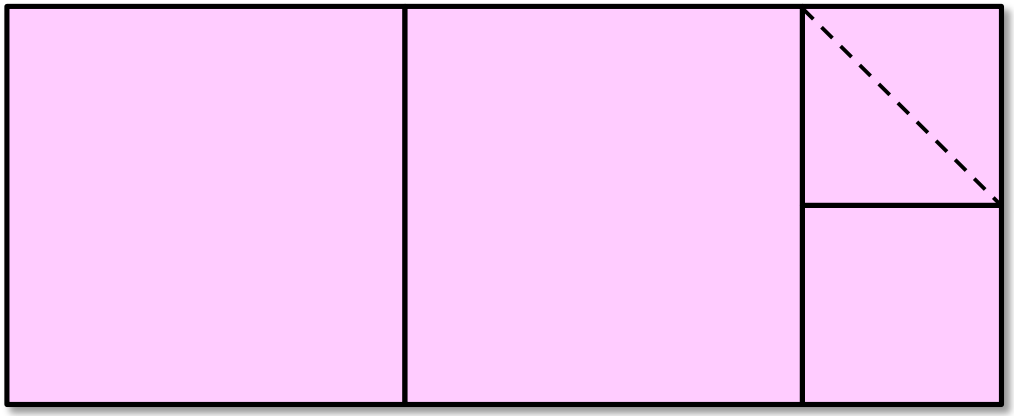
- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



- 長い辺を短い辺で割る.
- ぴったり割れないときは，余った長方形の長い辺を短い辺で割る.

解答

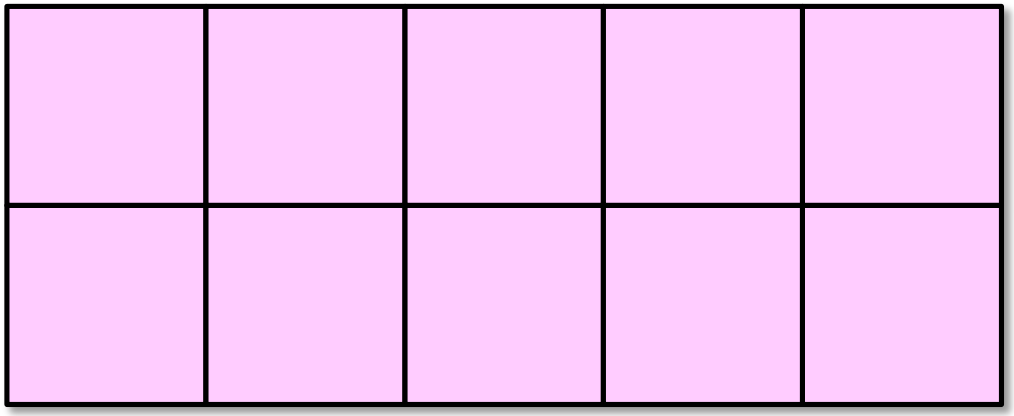
- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



- ぴったり割れたら，短い辺の長さが最大公約数.

解答

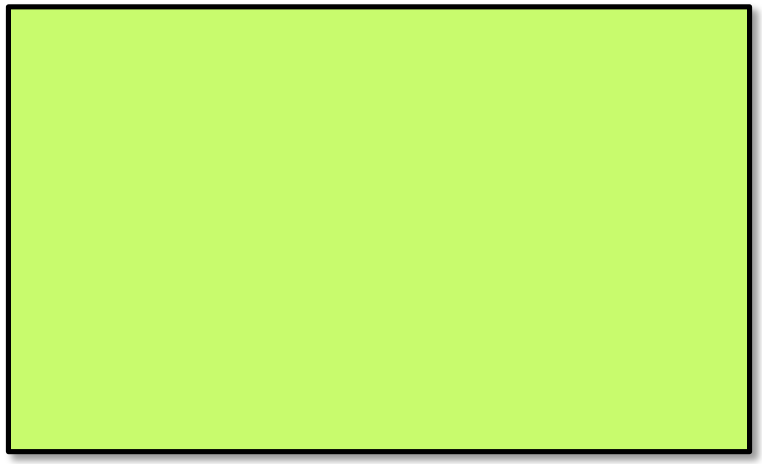
- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



- ぴったり割れたら，短い辺の長さが最大公約数.

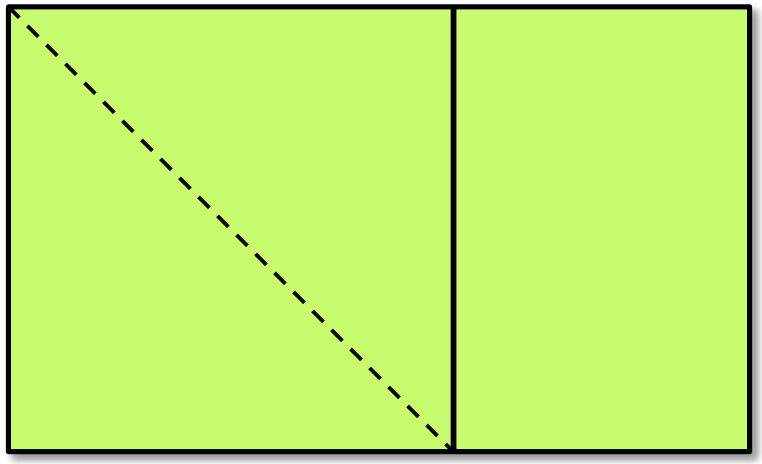
演習3

- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



解答

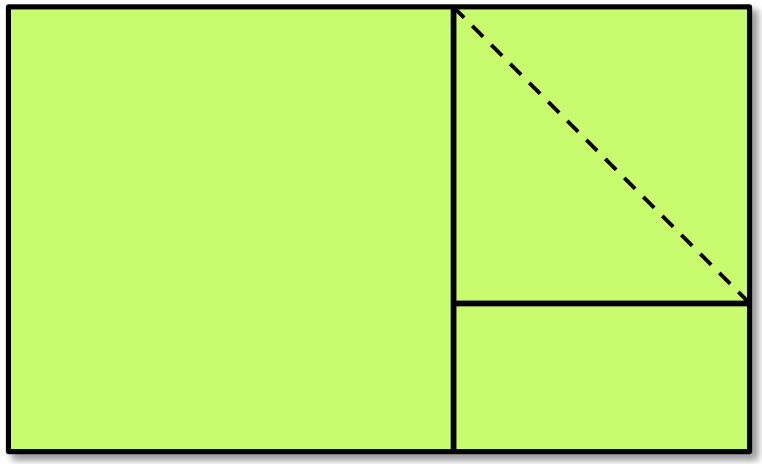
- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



- 長い辺を短い辺で割る.
- ぴったり割れないときは，余った長方形の長い辺を短い辺で割る.

解答

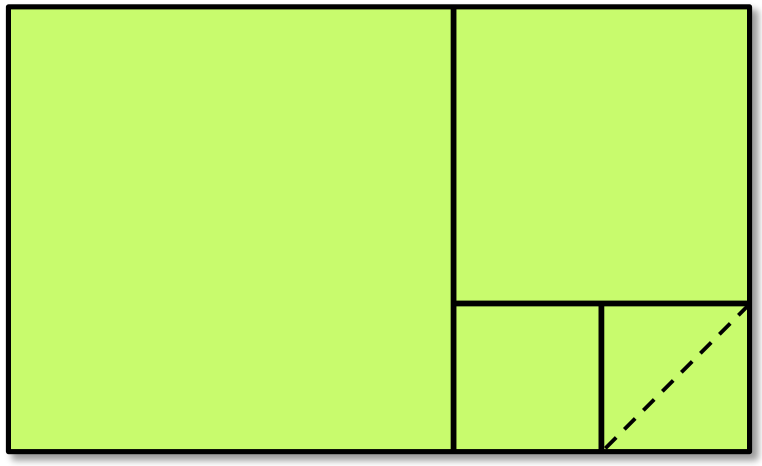
- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



- 長い辺を短い辺で割る.
- ぴったり割れないときは，余った長方形の長い辺を短い辺で割る.

解答

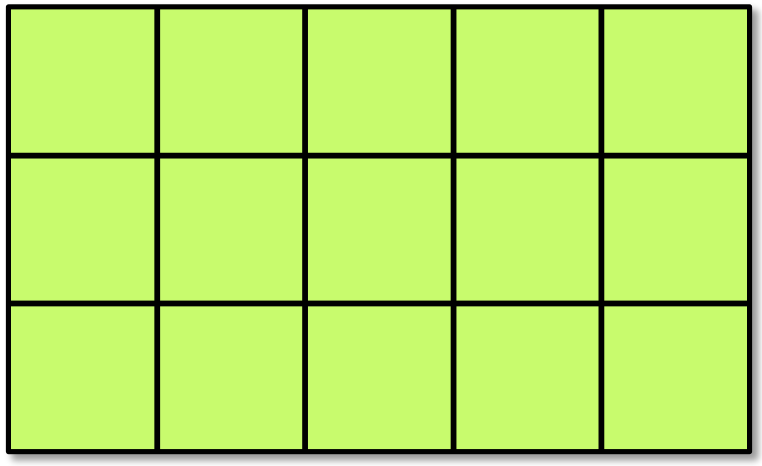
- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



- ぴったり割れたら，短い辺の長さが最大公約数.

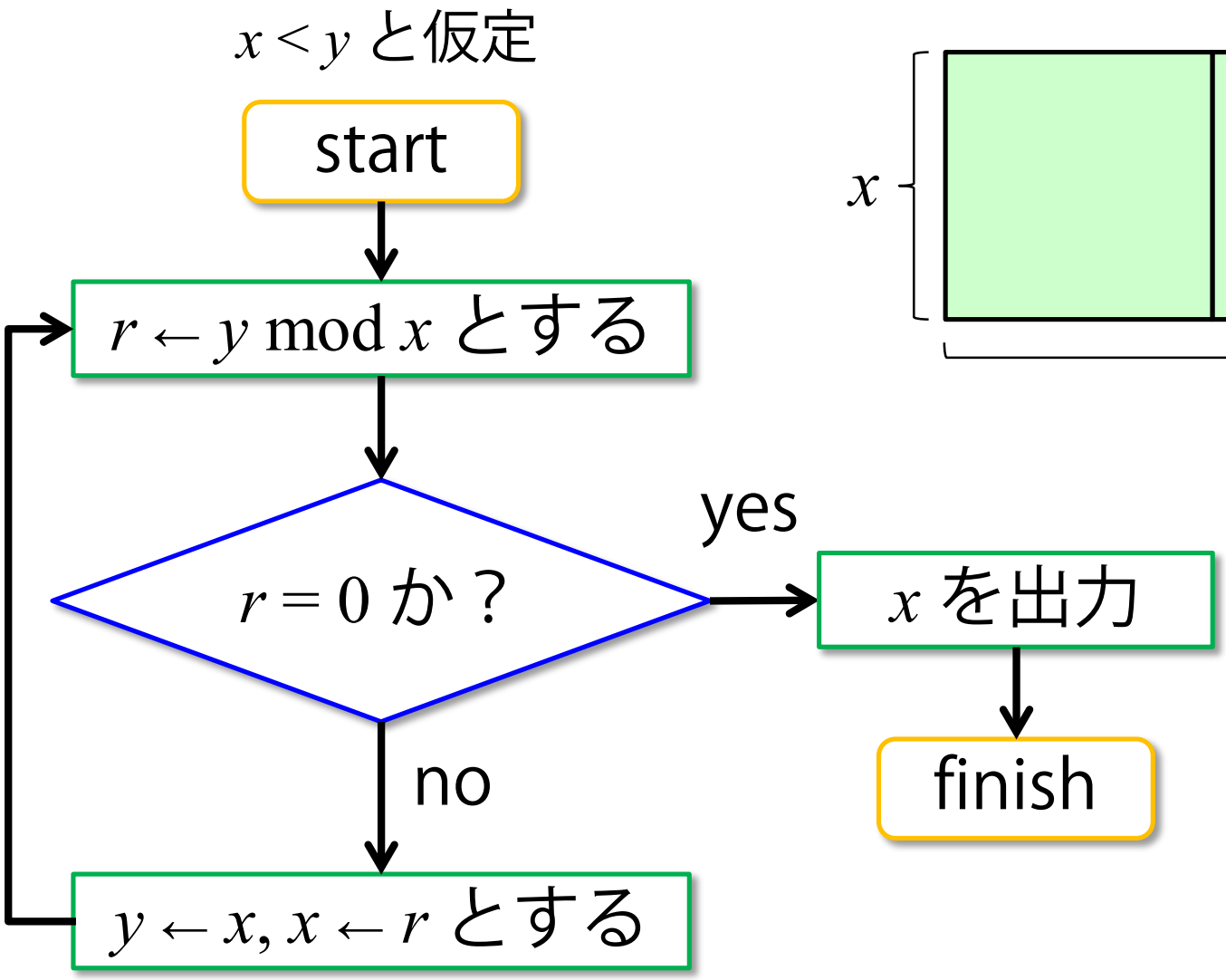
解答

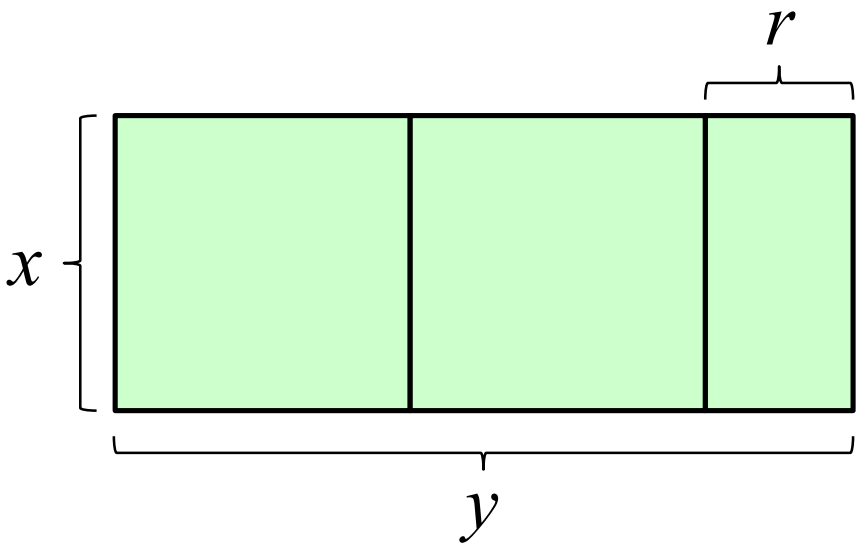
- 長方形の紙をなるべく大きな正方形に分割せよ.
- ただし，最初に定規で辺の長さを測ってはいけない.



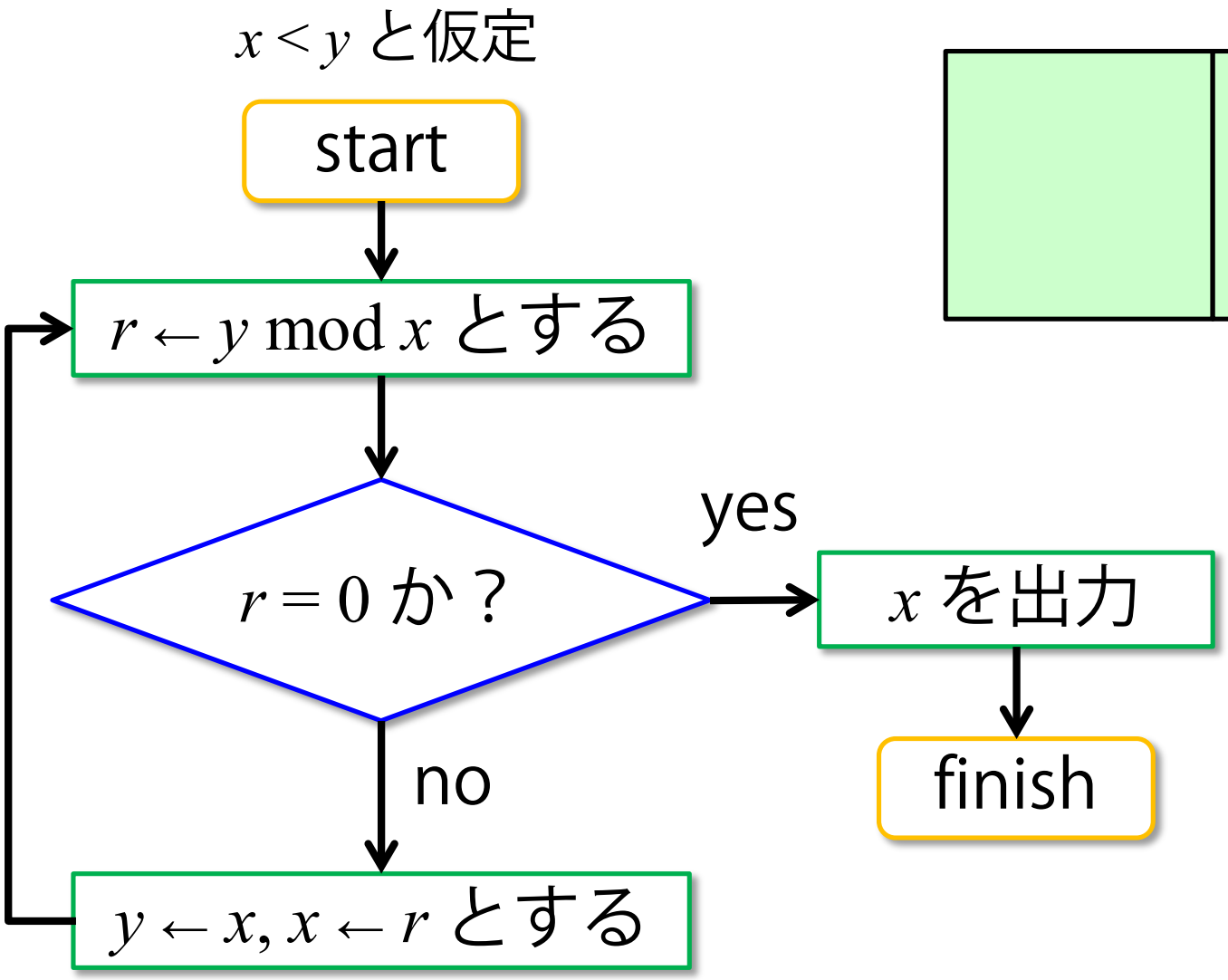
- ぴったり割れたら，短い辺の長さが最大公約数.

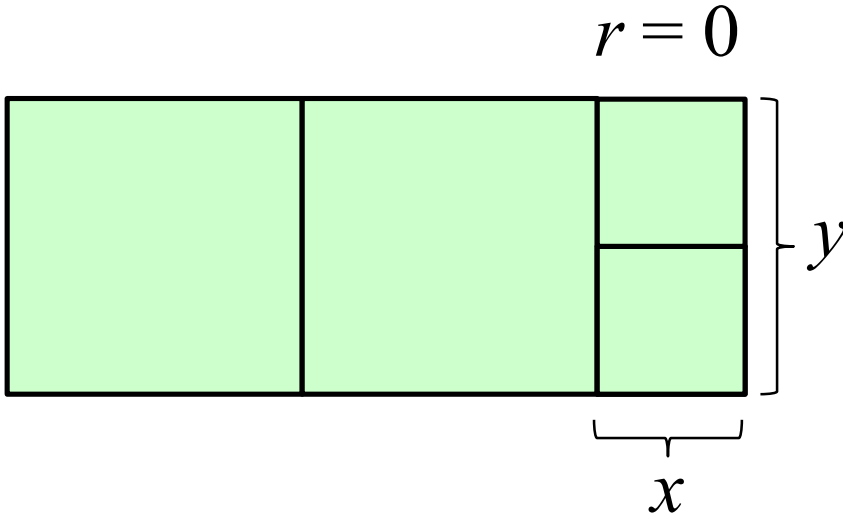
賢いアルゴリズム euclid_gcd



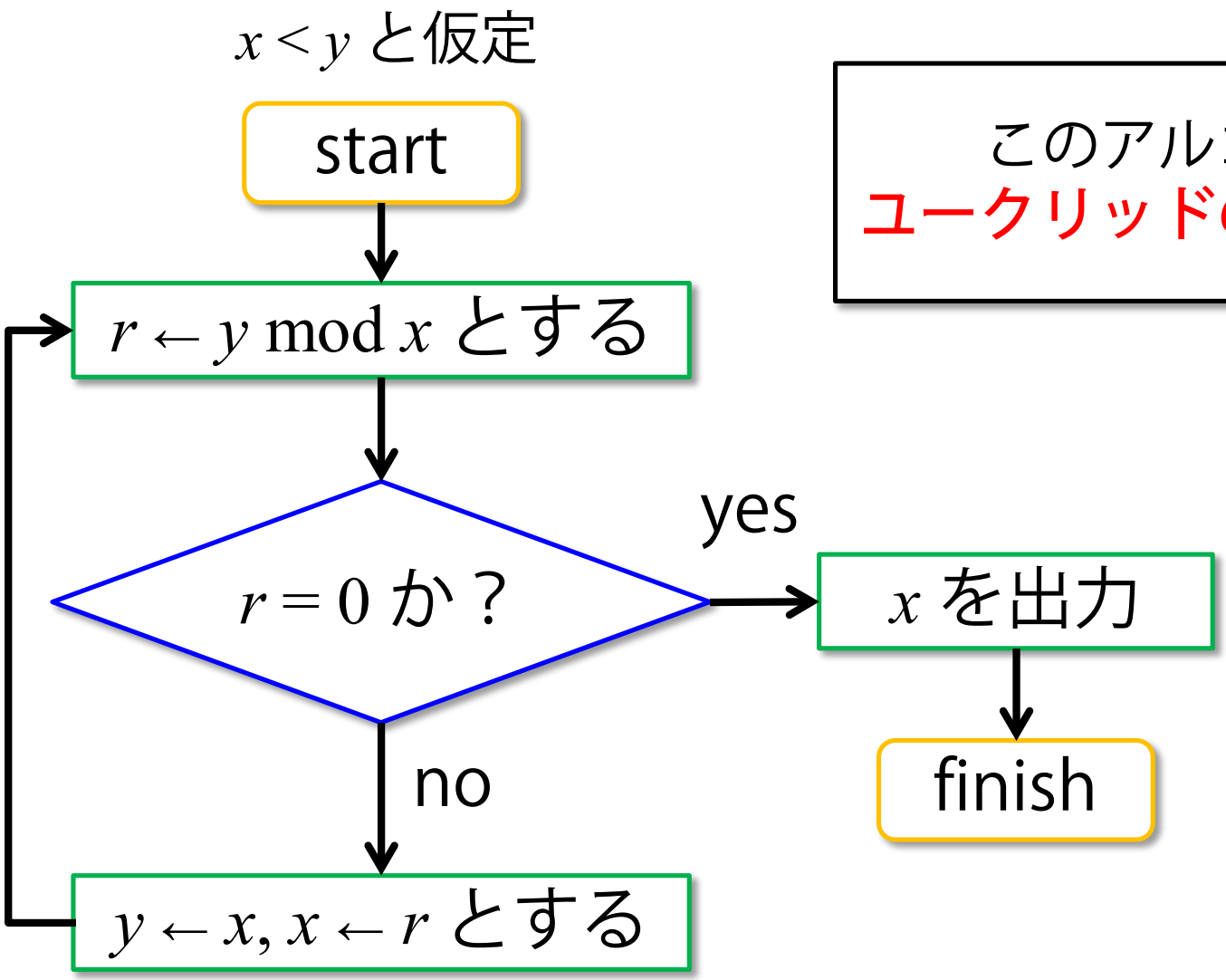


賢いアルゴリズム euclid_gcd





賢いアルゴリズム euclid_gcd



このアルゴリズムを
ユークリッドの互除法という

euclid_gcd の実行例

【問】 7289 と 740 の最大公約数を求めよ

- 7289 mod 740 = 629
- 740 mod 629 = 111
- 629 mod 111 = 74
- 111 mod 74 = 37
- 74 mod 37 = 0

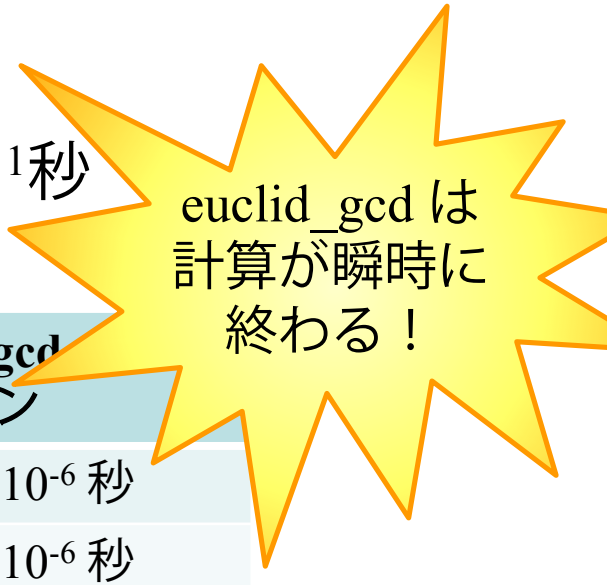
余りが 0 になったので,
37 = gcd(7289, 740)

euclid_gcd の計算時間見積もり

euclid_gcd の計算ステップ数 $\approx 2\log_2 n$

余りの計算 (mod) の1回の計算時間

– スーパーコンピュータ 1ステップ 10^{-11}秒



n	**naive_gcd** スパコン	**euclid_gcd** スパコン
1億	0.002 秒	0.56×10^{-6} 秒
1兆	20 秒	0.82×10^{-6} 秒
1京	55.5 時間	1.08×10^{-6} 秒
10^{50}	6.3×10^{31} 年	3.34×10^{-6} 秒
10^{100} ($\approx 2^{332}$)	6.3×10^{81} 年	6.66×10^{-6} 秒

おまけ：
因数分解よりもユークリッドの互除法

- x と y をそれぞれ因数分解すれば，
  最大公約数 $\gcd(x, y)$ は求まるが…

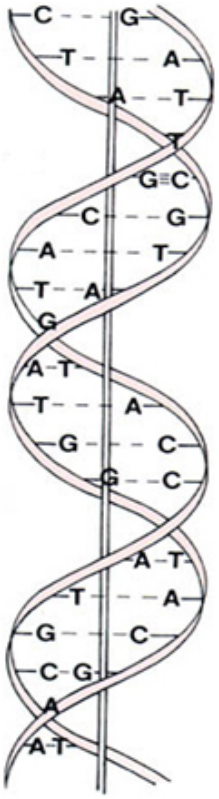
- 実は，大きな数の因数分解は難しい！
  （高速なアルゴリズムが知られていない）

- しかし，ユークリッドの互除法を用いれば，
  高速に $\gcd(x, y)$ を求めることができる^^

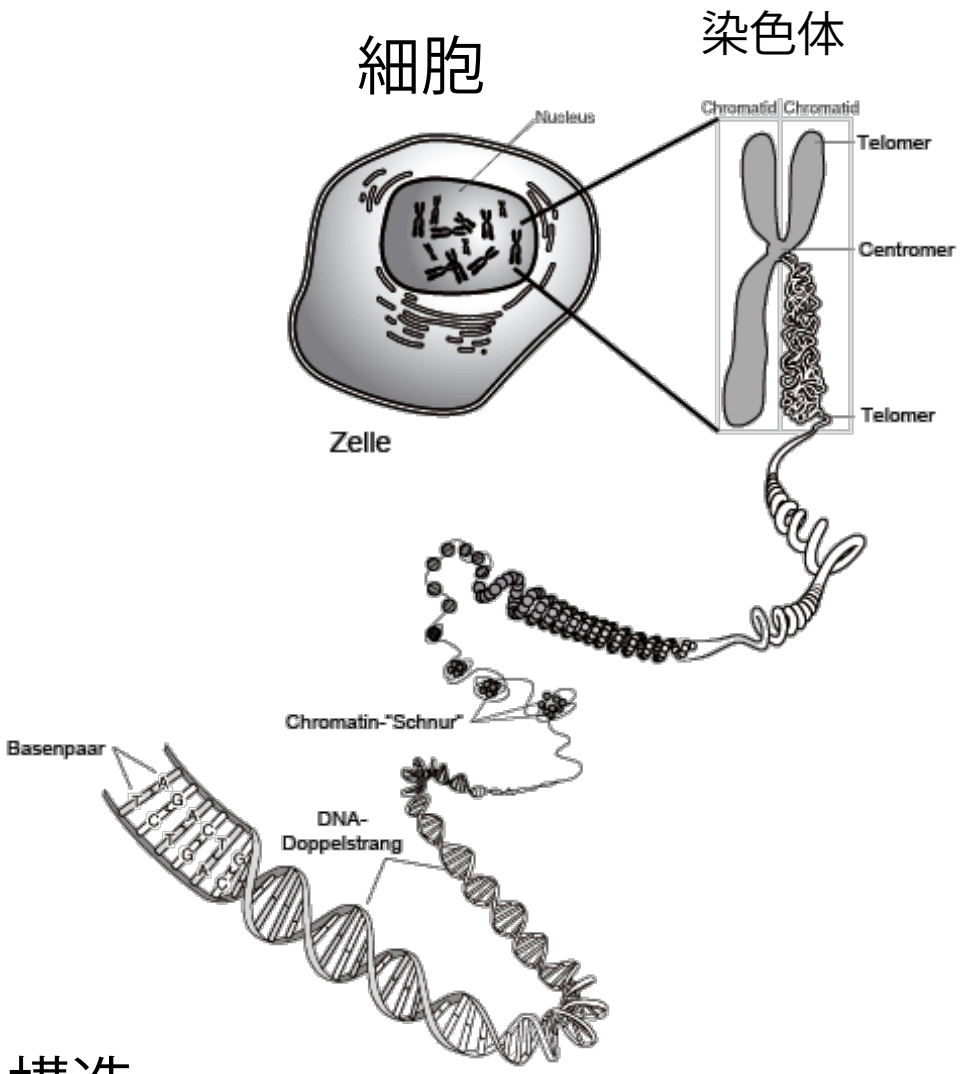
研究紹介

万物は文字列である

例１） DNA配列

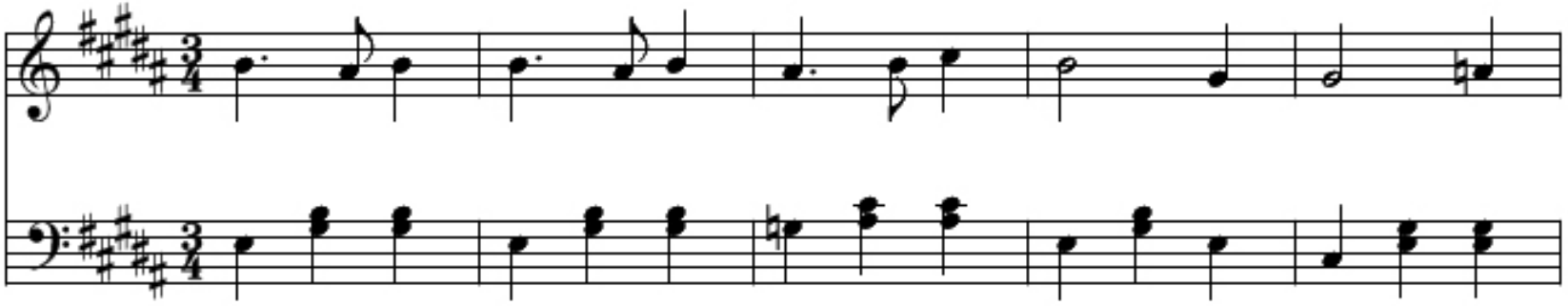


A, C, G, T からなる
２つの文字列のらせん構造



万物は文字列である

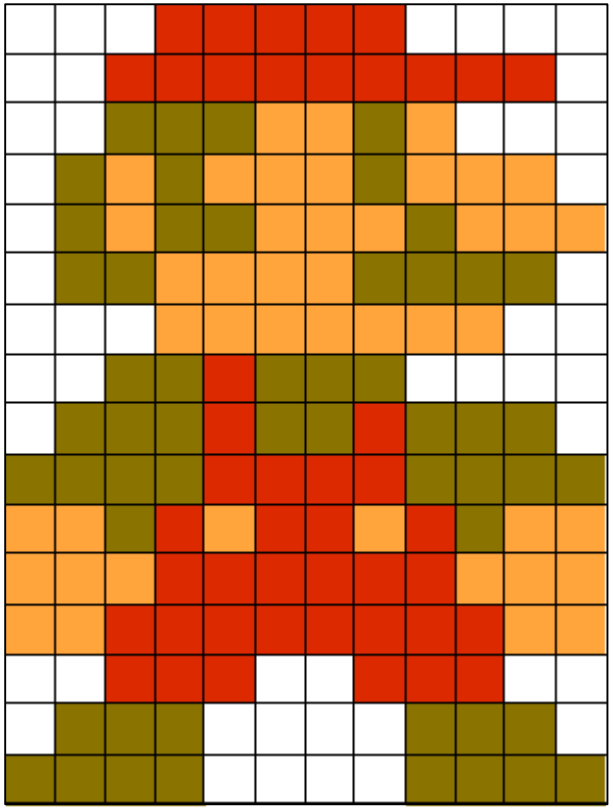
例２） 音楽データ

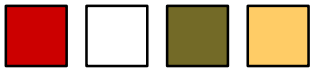


音符と休符の列

万物は文字列である

例３） 画像データ

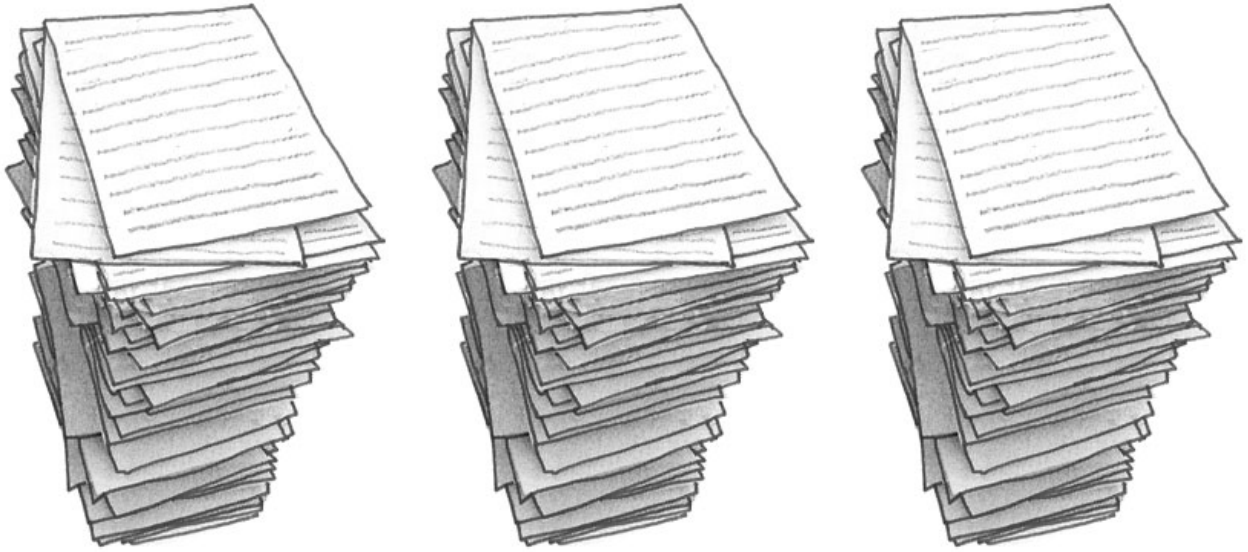




色を記号だとみなせば
画像も文字列である

コンピュータ上のデータ＝文字列

- コンピュータ上のデータは文字列である
- 大量に溢れる文字列データをいかに効率よく処理するかがカギ



データ圧縮

文字列データを圧縮して表現すれば、
データの「無駄」を削減できる

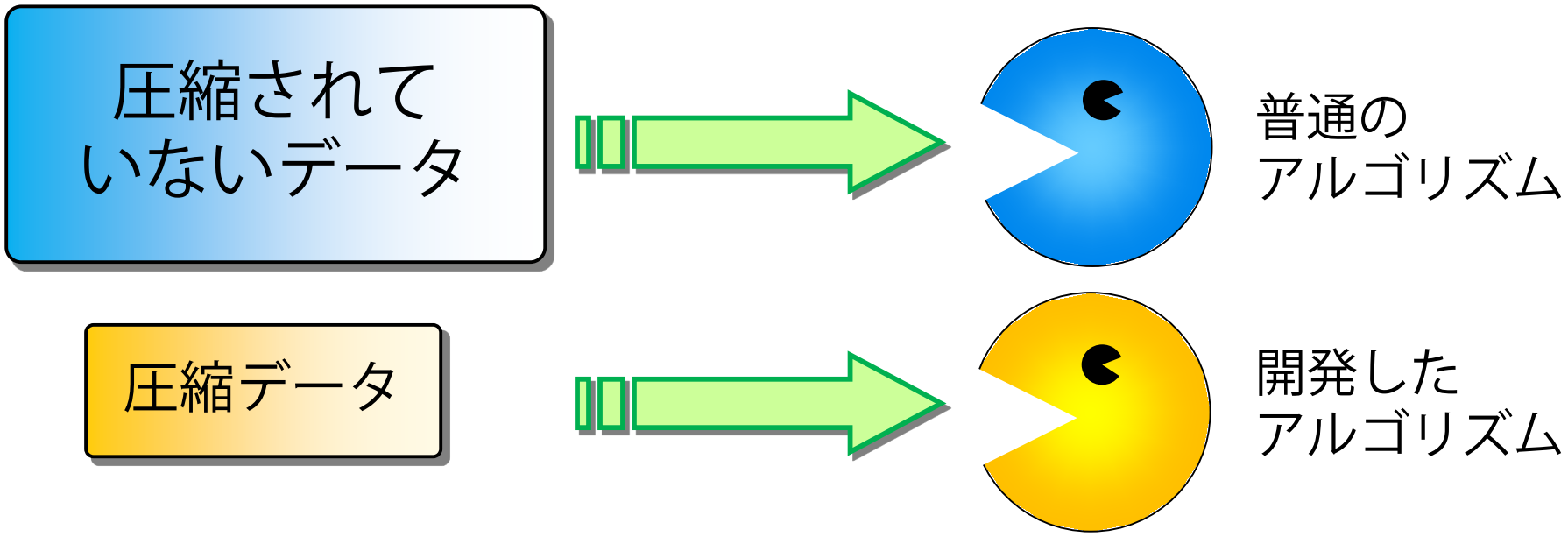
例１）すももももももものうち
　　＝すも^{8}のうち

例２）にわにわにわにわとりがいる
　　＝(にわ)^{4}とりがいる

研究成果

圧縮された文字列データ上で
情報検索を高速に行うアルゴリズムを開発した

- 大量のデータも圧縮すれば小さくなる
- 圧縮することで、情報検索の速度も向上！



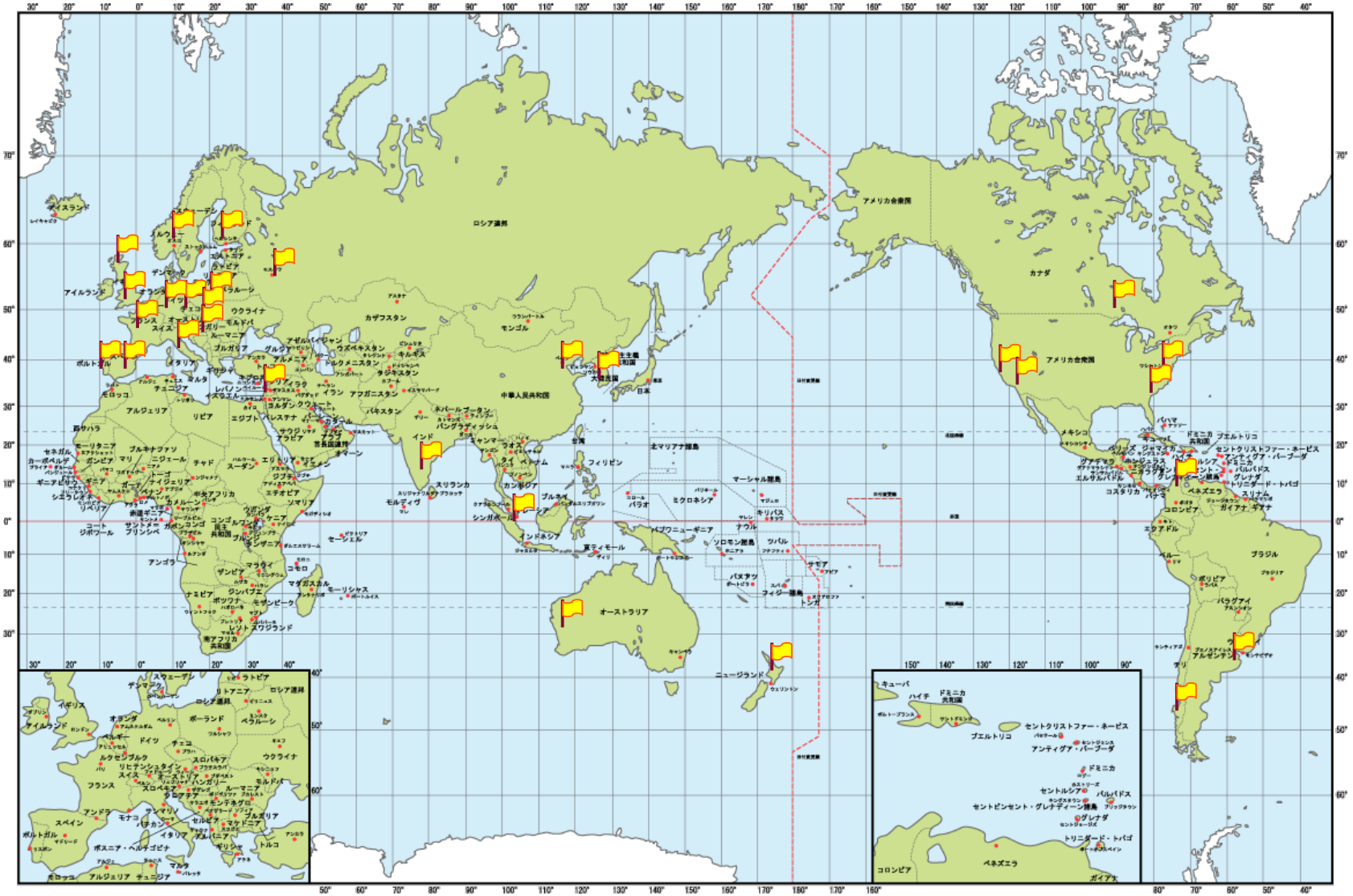
研究成果の発表の場＝世界

- 国際会議に参加して、自分たちの研究成果を世界中の研究者に報告する（海外出張）

- これまでに海外出張で行った国々：

  イギリス，フィンランド，ノルウェー，
  ポーランド，チェコ，スロバキア，ドイツ，
  フランス，スペイン，イタリア，ポルトガル，
  ロシア，ハンガリー，オーストリア，イスラエル，
  アメリカ，カナダ，チリ，アルゼンチン，
  コロンビア，オーストラリア，ニュージーランド，
  インド，シンガポール，韓国，中国

世界地図に 🚩 を立ててみる



情報科学の歴史

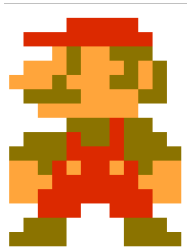
1822年
- 階差機関
  by Babbage

1936年
- チューリング機械
  by Turing

1950年代
- 情報理論
  by Shannon
- プログラム言語
  (Fortran, Lips, Algol, Cobol, ...)

1960-1970年代
- ワークステーション,
  データベース
- ワードプロセッサー

1980年代
- パソコン
- オブジェクト指向言語
- コンピュータゲーム



1990年代
- インターネット
- 携帯電話

2000年代
- 電子マネー
- ユビキタス
- モバイル (iPod, iPhone)

2010年代
- 3D-TV, iPad,
  smart phone

情報科学の歴史

1822年
- 階差機関
  by Babbage

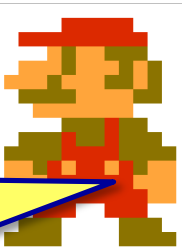
1936年
- チューリング機
  by Turing

1950年代
- 情報理論
  by Shann
-
  (Fortra

1960-1970年代
- ワ
  データベース
- ワードプロセッ

1980年代
- ン
  ジェクト指向言語
  ピュ ーム
  ル (iPod, iPhone)

0年代
- 3D-TV, iPad,
  smart phone



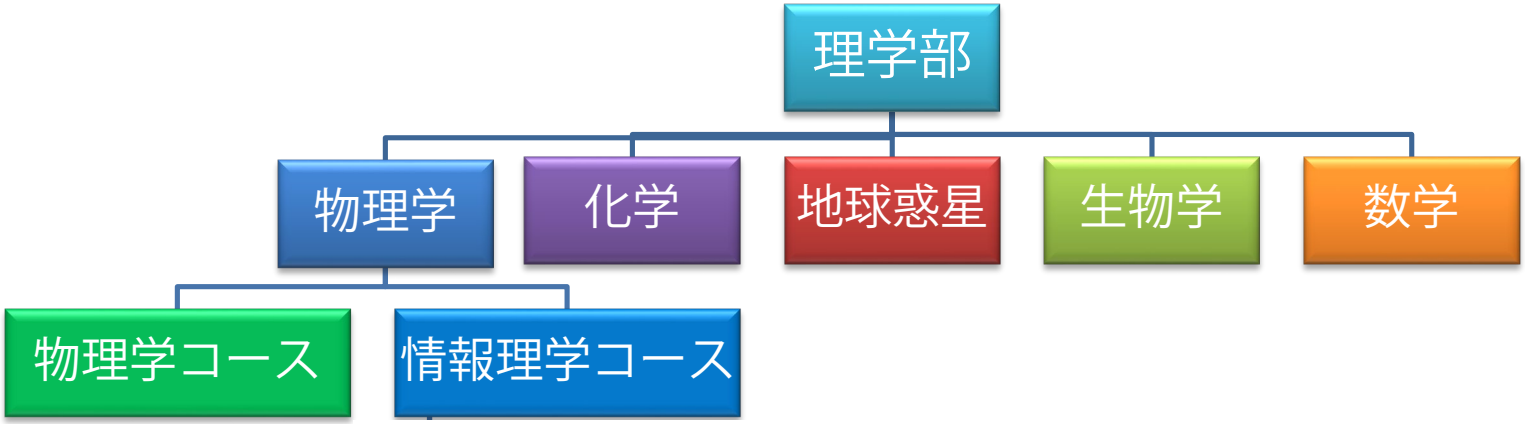
- 情報科学は非常に若い分野
- 学生や若手研究者が世界で
  活躍する余地が十二分にある

九州大学の情報系学科

九州大学 工学部 電気情報工学科

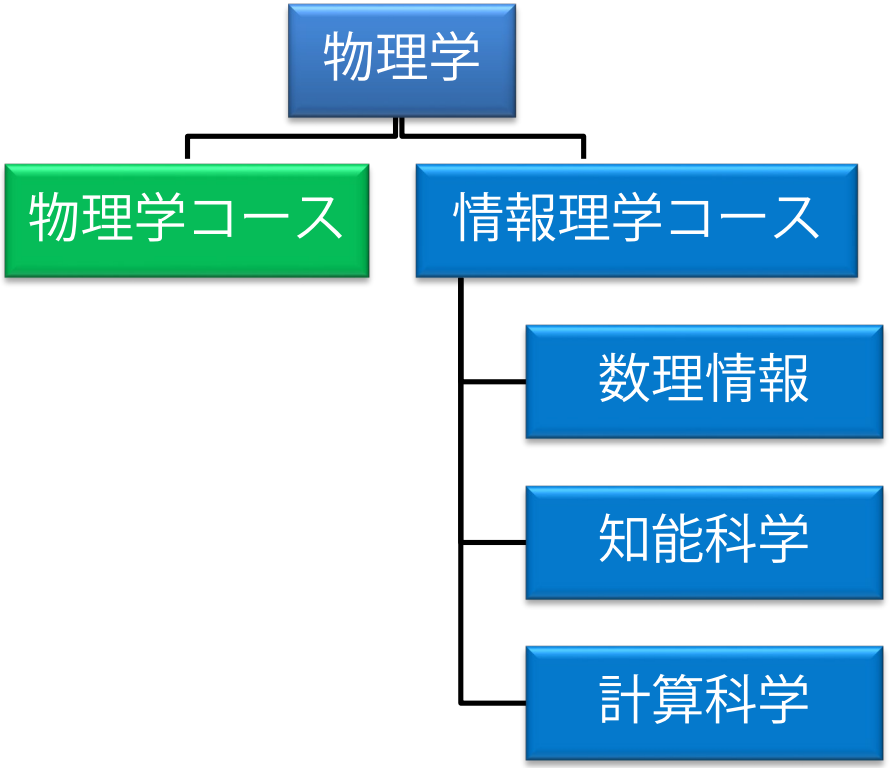
２年生前期に
課程を選択

九州大学 理学部 物理学科



２年生前期に
コースを選択

九州大学 理学部 物理学科



アルゴリズム，暗号理論など

人工知能，計算学習理論など

計算機シミュレーションなど

まとめ

まとめ

- 何かに興味を持ったら、とことんやってみよう
  - ゲームがきっかけで、准教授になることもある
- 情報科学は比較的新しい学問分野
  - 若者が活躍するための余地が十分に残されている
  - 研究成果を残せば、海外~~旅行~~出張に行ける！
- 九州大学で情報科学について学ぶなら，
  工学部 電気情報工学科 **または**
  理学部 物理学科 情報理学コース