アルゴリズム論
（2015年度後期）

岩手県立大学
Iwate Prefectural University

佐々木研（情報システム構築学講座）
講師　山田敬三
k-yamada@iwate-pu.ac.jp

アルゴリズム論
（第1回）

岩手県立大学
Iwate Prefectural University

佐々木研（情報システム構築学講座）
講師　山田敬三
k-yamada@iwate-pu.ac.jp

データ構造とアルゴリズム

岩手県立大学
Iwate Prefectural University

データ構造

データ-こうぞう【ー構造】

〔data structure〕
情報が表す概念に従って規定される構造。

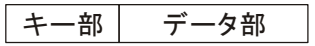
三省堂提供「大辞林 第二版」

データ構造

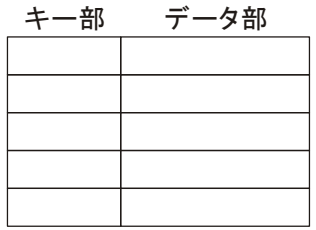
- 使用するデータ群を効率よく操作するために使用するデータの構造

- 例）
  配列，線形連結リスト，スタック，キュー，グラフ，木，ハッシュテーブル，レコード，etc.

レコード



(a) レコード



(b) テーブル

- キー
  - データを識別するための情報
- レコード
  - キーとデータの組
- テーブル
  - レコードの集合

アルゴリズム

アルゴリズム【algorithm】

計算や問題を解決するための手順・方式。

〔もとは算用数字を用いた筆算のこと。アラビアの数学者アル=フワリズミの名にちなむ〕

三省堂提供「デイリー 新語辞典」

アルゴリズム

- 何らかの問題に対して，コンピュータを用いて解決するための方法や計算手順，手続き

- プログラム　＝　アルゴリズム　＋　データ

データ構造とアルゴリズム

- アルゴリズム
  - コンピュータを用いて解決するための方法や計算手順，手続き
- データ構造
  - 使用するデータ群を効率よく操作するために使用するデータの構造

例

- 1からnまでの総和を求める.

$$\sum_{k=1}^{n} k = 1 + 2 + \cdots + (n-1) + n$$

- これをプログラムにすると？(n=10)

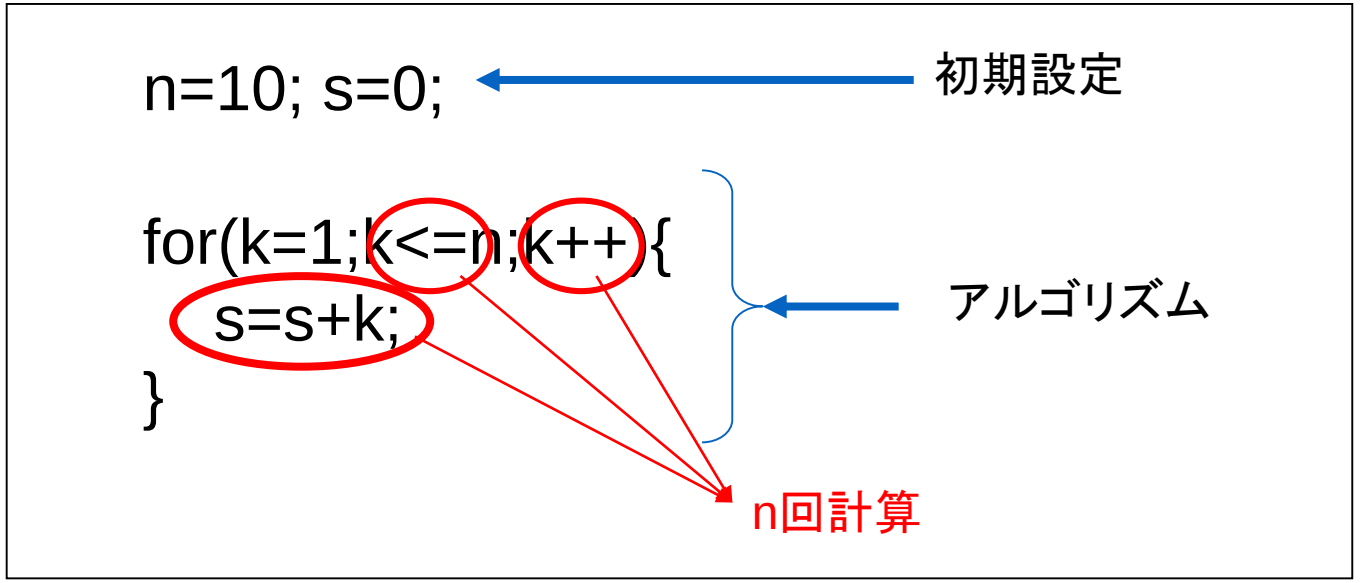
```
n=10; s=0;

for(k=1;k<=n;k++){
    s=s+k;
}
```

初期設定

アルゴリズム

アルゴリズム1

- これをプログラムにすると？



- どのくらい時間がかかるのか？

これって、効率良いの？

アルゴリズム２

- 16世紀ガウスが考えた

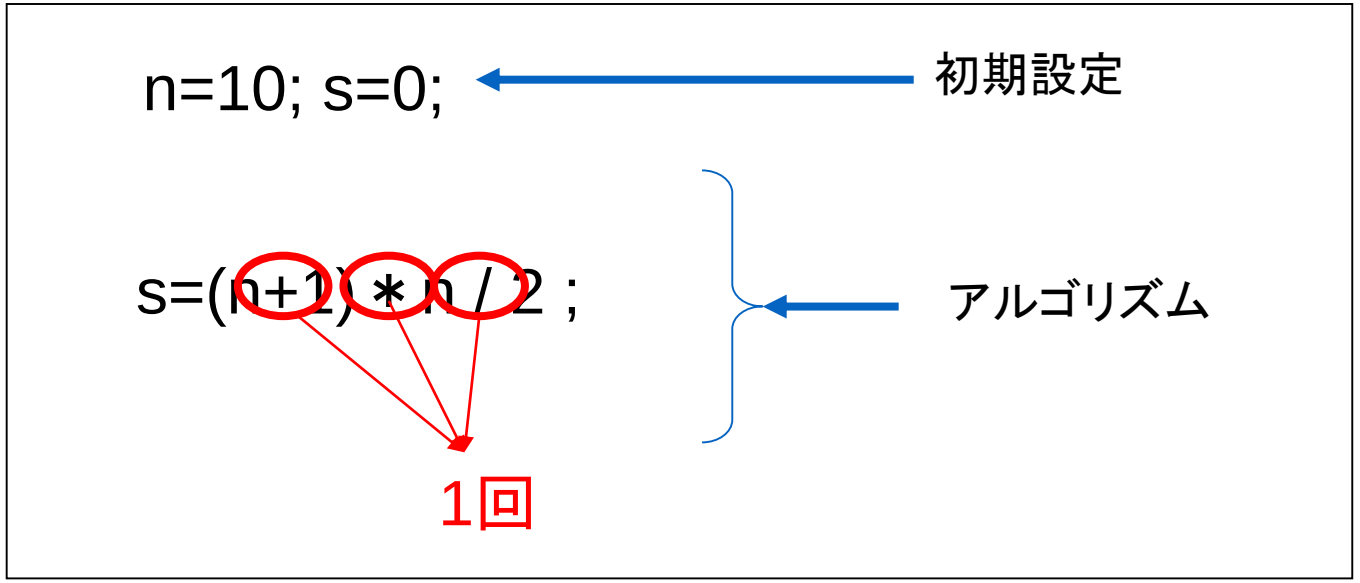
$$
\begin{array}{ccccccccc}
1 & + & 2 & + & 3 & + \cdots + & (n-1) & + & n \\
+\ n & + & (n-1) & + & (n-2) & + \cdots + & 2 & + & 1 \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
(n+1) & + & (n+1) & + & (n+1) & + \cdots + & (n+1) & + & (n+1)
\end{array}
$$

n個

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$$

アルゴリズム2

- これをプログラムにすると



計算時間

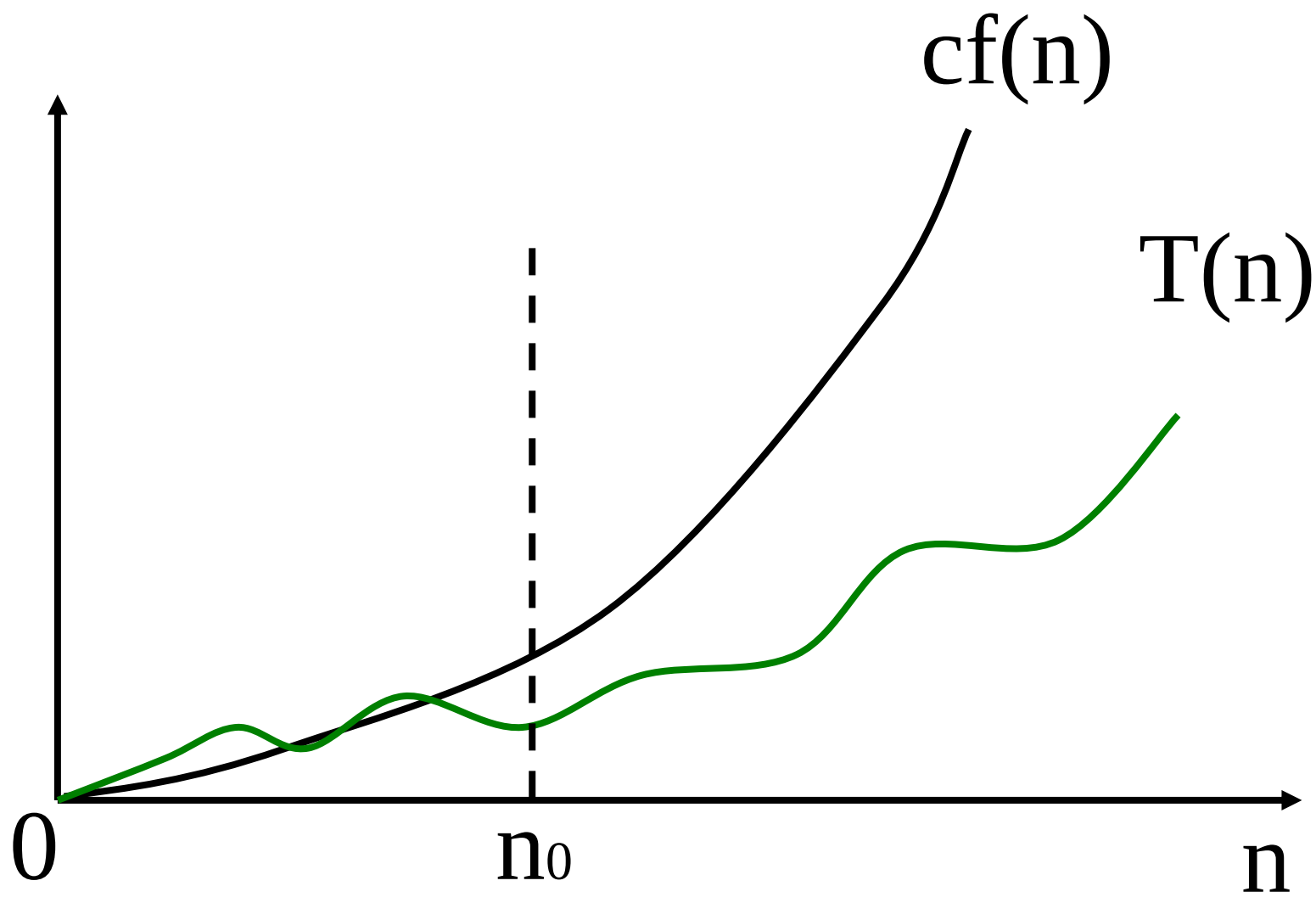
- 2つのアルゴリズムを評価
  - 計算時間 T(n) で比較
- アルゴリズム１
  - T（n)=3n
    - n回計算するところが3回
    - 計算時間はnに依存
- アルゴリズム２
  - T(n)=3
    - nに依存しない

計算量

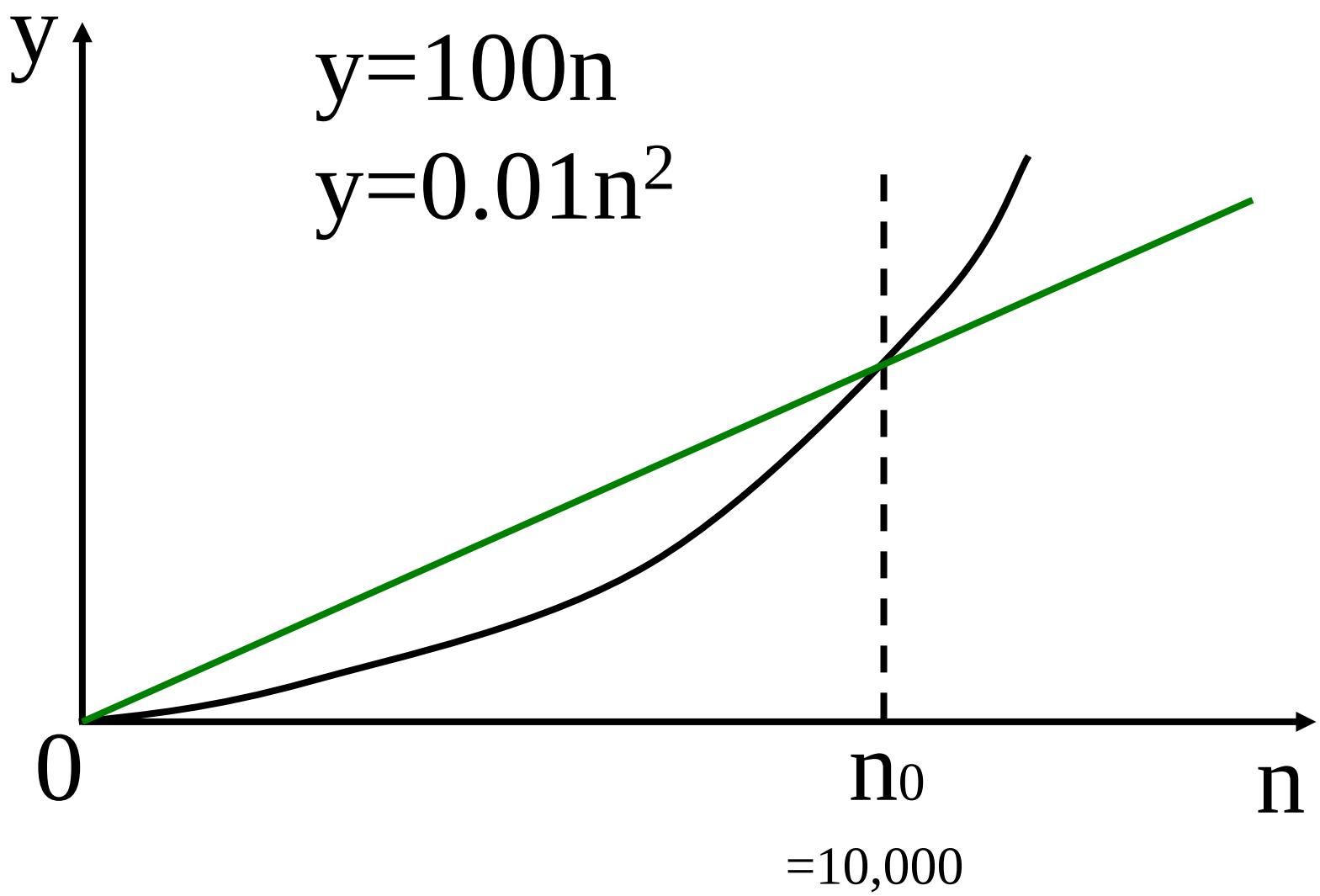
- 漸近的計算量で評価する
- $T(n) = O\big(f(n)\big)$ とは、

  $\exists c, n_0\ \forall n \geq n_0$ ならば $T(n) \leq cf(n)$

計算量



nとn²との違い



計算量

- 漸近的計算量で評価
  - O(n):オーダ
  - T(n)＝3n　→　O(n)
  - T(n)=3　→　O(1)

おおまかな量だけを捉える.

例：$T(2n^3+5n^2+n)=O(n^3)$

漸近的計算量のアイディア

- 入力が一桁増えると、計算時間が何桁増えるか

- 「一桁増える」感覚は？
  - 「なわとび」、「まりつき」で考える

漸近的計算量の例

- $O(1)$
- $O(\log n)$
- $O(n)$
- $O(n^2)$
- $O(n^3)$
- $O(2^n)$

計算量

問題空間 N に対して，定数 A とすると

$$1 < \log_A N < N < N \log_A N < N^A < A^N$$

A= 2

N	1	logN	N^A	A^N
2	1	1	4	4
4	1	2	16	16
8	1	3	64	256
16	1	4	256	65,536
100	1	7	10,000	1.27E+30
1,000	1	10	1,000,000	1.07E+301

処理性能

- 高性能なアルゴリズム
  - 計算量が少ない.
    - 少なければ少ないほどよい.
- 計算機で可能な処理
  - 高々多項式時間
  - 指数時間の処理は現実的に無理
    - 計算にかかる時間が天文学的になる

この講義の流れ

1. 問題の定義(何を解くか)
2. データ構造の定義
3. 解法(アルゴリズム)の解説
4. 解法の評価（計算量など）