

# 計算量について

---

佐藤謙成

May 24, 2025

KUT-PG

## ① 計算量 (オーダー)

## 計算量 (オーダー)

---

# オーダー記法

## オーダー記法とは

アルゴリズムの計算量を評価するための記法.  $t$  を時間とすると,  $O(t)$  で表すことができる.  $t$  は小さい方が計算量が少ないため, 評価は高くなる.

- アルゴリズムによって所要時間が異なる
- 良いアルゴリズムの場合は解決したい問題の規模が大きくなっても対抗可能である.

以下は,  $t$  に入る代表格である.

$$\log n < \sqrt{n} < n < n \log n < n^2 < n^3 < \dots < 2^n < n!$$

できるだけ  $\log n$  になるようにアルゴリズムを組んでいきたい.

## オーダー記法 ( $t$ ) の決め方

- ➊ アルゴリズムの時間計算量を入力サイズ  $n$  を用いた関数で表す.
- ➋ その関数の中で主要項<sup>a</sup>を見つける.
- ➌ 主要項の係数を削除した関数が  $f(n)$  である時, アルゴリズムの時間計算量は  $O(f(n))$  であるという.

---

<sup>a</sup> $n$  が一番大きいもの

# 実行時間

アルゴリズムの実行時間に関して、以下の語句で表現される。

- ① 時間計算量
- ② 最良時間計算量
- ③ 最悪時間計算量
- ④ 領域計算量
- ⑤ 平均時間計算量

## 最良時間計算量

最も速くそのアルゴリズムを実行できる入力の世界時間計算量。

## 最悪時間計算量

最も時間のかかる入力に対する時間計算量。

## 平均時間計算量

入力データがランダムに分布している場合の計算時間を示す。

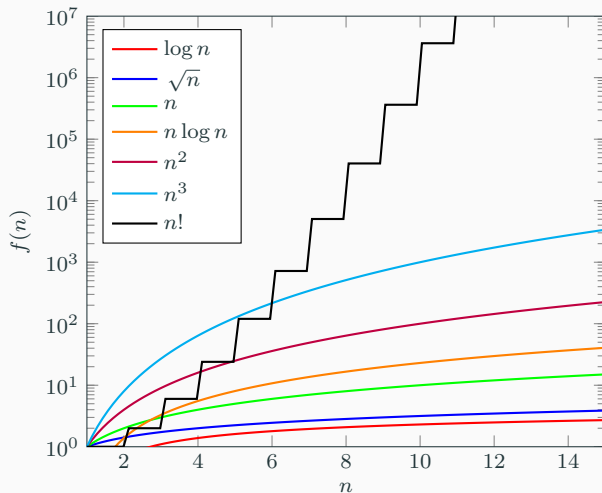
## 領域計算量

アルゴリズムを実行する際にどれだけのメモリ (領域) が必要かを表す。

# 計算量の評価

計算量の評価は以下の通りとなる.

$$\log n < \sqrt{n} < n < n \log n < n^2 < n^3 < \dots < 2^n < n!$$



## 基本演算での時間計算量

- 算術演算
- 論理演算
- 入出力の命令

これらの基本演算の時間計算量は  $O(1)$ (定数時間) となる.

## 時間計算量の例.1

以下のアルゴリズムの時間計算量を求めていく.

### アルゴリズム A

$$10n^2 + 100n + 10000$$

### アルゴリズム B

$$n^4 - n^3 + 100$$

### 再掲

- ① アルゴリズムの時間計算量を入力サイズ  $n$  を用いた関数で表す.
- ② その関数の中で主要項<sup>a</sup>を見つける.
- ③ 主要項の係数を削除した関数が  $f(n)$  である時, アルゴリズムの時間計算量は  $O(f(n))$  であるという.

---

<sup>a</sup> $n$  が一番大きいもの



## 時間計算量の例.2

以下のコードを元に時間計算量を求めていきたい。(計算量オーダーは、最悪時間計算量について考えるのが基本である.)

```
1      for(int i = 1; i < n; i++) {  
2          for(int j = i + 1; j < n + 1; j++) {  
3              if(x[i] == x[j]) {  
4                  cout << "x[i]とx[j]は同じである.";  
5              }  
6          }  
7      }
```

- if 文の時間計算量は、 $O(1)$  である.
- 外側の for 文の繰り返し回数は、 $n - 1$  回.
- 内側の for 文の繰り返し回数は、 $n - i$  回.

このアルゴリズムの時間計算量は、

$$O(1) \times \frac{n(n-1)}{2} = O(n^2)$$

## 計算量の使い方

実際の問題をに対してアルゴリズムを設計する時の考え方について考えていく.

通常の計算機では 1 秒間に処理できる for 文のループの回数は,  $10^8$  回程度である. (Atcoder では通常, 制限時間が 2 秒間であるので for 文のループの回数は  $2 \times 10^8$  回程度である.)

$\log n$	$n$	$n \log n$	$n^2$	$n^3$	$2^n$	$n!$
2	5	12	25	130	30	120
3	10	33	100	1000	1024	362880
4	15	59	225	3375	32768	—
4	20	86	400	8000	1048576	—
5	25	116	625	15625	—	—
5	30	147	900	27000	—	—
7	100	664	10000	1000000	—	—