

## 5.3 計算量

世の中には、以下のように解ける問題と解けない問題が存在します。

- **解ける問題**：問題を解決するためのアルゴリズムが存在する。
- **解けない問題**：問題を解決するためのアルゴリズムが存在しない (見つかっていない)。

なお、解ける問題の中には、その答えを見出すのが易しいものから困難なものまで様々です。困難なものとしては、

- 無理数の真の数値 ( $\pi = 3.14\dots$ ,  $e = 2.71\dots$ ,  $\sqrt{2} = 1.41\dots$  など)
- 組み合わせ数が非常に多い組み合わせ問題 (ハノイの塔, 巡回セールスマンなど)
- 非常に大きな2つの素数の積の因数分解 (公開鍵暗号の安全性を保障)

などがあります。これらの問題は、人間であろうがコンピュータであろうがその答えを導くのは非常に困難です (現時点では無理)。しかしながら、コンピュータが大量のデータを高速かつ正確に処理できるという利点を生かして、ある程度までの答えを見出すことが有意義で意味ある場合も数多く存在します。

情報科学の分野では、問題を解決するためのアルゴリズムをコンピュータで処理させようとするとき、その処理に必要な資源 (計算時間, 記憶容量など) を**計算量** (complexity) という抽象的な尺度で評価します。計算量には、**アルゴリズムの実行にどれだけ時間がかかるか**という尺度に基づいて評価する**時間計算量** (time complexity) と、**アルゴリズムの実行にどれだけ記憶領域 (メモリ) が必要か**という尺度に基づいて評価する**領域計算量** (space complexity) があります。通常は、単に計算量といえば時間計算量のことを指し、処理に必要な計算時間が評価の中心となります<sup>7</sup>。

また、計算量は最悪の入力データを想定して評価します。これを**最大時間計算量** (worst case time complexity) と呼びます。これに対して、全ての入力データに対する計算量の平均値を**平均時間計算量** (average case time complexity) と呼びます。一般には、計算量といえば最大時間計算量のことを指しますが、入力データの良し悪しによって計算量の評価が左右されるような場合は平均時間計算量が重要な意味を持ちます。

計算量を求めるには、 $n$  個の入力データに対してアルゴリズムの処理に必要な時間を算出します。より詳しく述べれば、アルゴリズムの中で実行される

- 比較演算
- 四則演算

などの各演算の回数をカウントし、各演算を1回実行するのに必要な時間を掛け、その総和を求めます。例えば、1個の入力データの処理に0.001秒かかる演算が1000回必要なとき、 $n$  個のデータを処理するには  $0.001 \times 1000 \times n (= n)$  秒の処理時間が必要となります。もう1つの例として、

<sup>7</sup>領域計算量も時間計算量と同様の評価ができます。ただし、計算時間が増えることに比べれば、コンピュータの記憶領域は比較的小さく限りがあるため、大量にデータを扱う場合は記憶領域を節約するようなアルゴリズムが好まれます (または、記憶領域をなるべく節約するようにプログラムを設計します)。これを時間と空間のトレードオフといい、領域計算量は時間計算量に転換されて評価されます。

1個の入力データの処理に0.001秒かかる演算が10000回と0.005秒かかる演算が200回必要なとき、 $n$ 個のデータを処理するには $0.001 \times 10000 \times n + 0.005 \times 200 \times n \times n (= n^2 + 10n)$ 秒の処理時間が必要となります。

さらに、計算量には**オーダー記法**という $n$ の値が十分大きいところで計算量を漸近的に評価する手段があります。オーダー記法を用いると、最初の例は $O(n)$ 、2つ目の例は $O(n^2)$ と表記することができます( $O$ はオーダー(Order)の頭文字)。ただし、2つ目の例が $O(n^2)$ と表記されているのは、 $n$ が非常に大きくなると $n^2$ に比べて $10n$ は非常に小さくなり無視されるためです(これを計算量の第1次近似といいます)。

最後に、アルゴリズムの計算量による評価について述べておきます。大きく分けてアルゴリズムには、 $O(n)$ 、 $O(n \log n)$ 、 $O(n^2)$ 、 $O(n^3)$ のように $n$ の多項式のオーダーとなる**多項式時間アルゴリズム**(polynomial time algorithm)と、 $O(2^n)$ 、 $O(3^n)$ 、 $O(n!)$ 、 $O(n^n)$ のように $n$ の指数関数のオーダーとなる**指数時間アルゴリズム**(exponential time algorithm)があります。前者は、 $n$ が多少大きくなっても計算量がそれほど大きくならないのに対して、後者は、 $n$ が少し増えただけで爆発的に計算量が大きくなります(表5.2; 1秒間に $10^9$ 回(クロック数が1GHz)の演算能力を持つコンピュータを使用したものとして算出)。従って、多項式時間アルゴリズムはコンピュータに適したアルゴリズムといえ、指数時間アルゴリズムはコンピュータには適さないといえます。

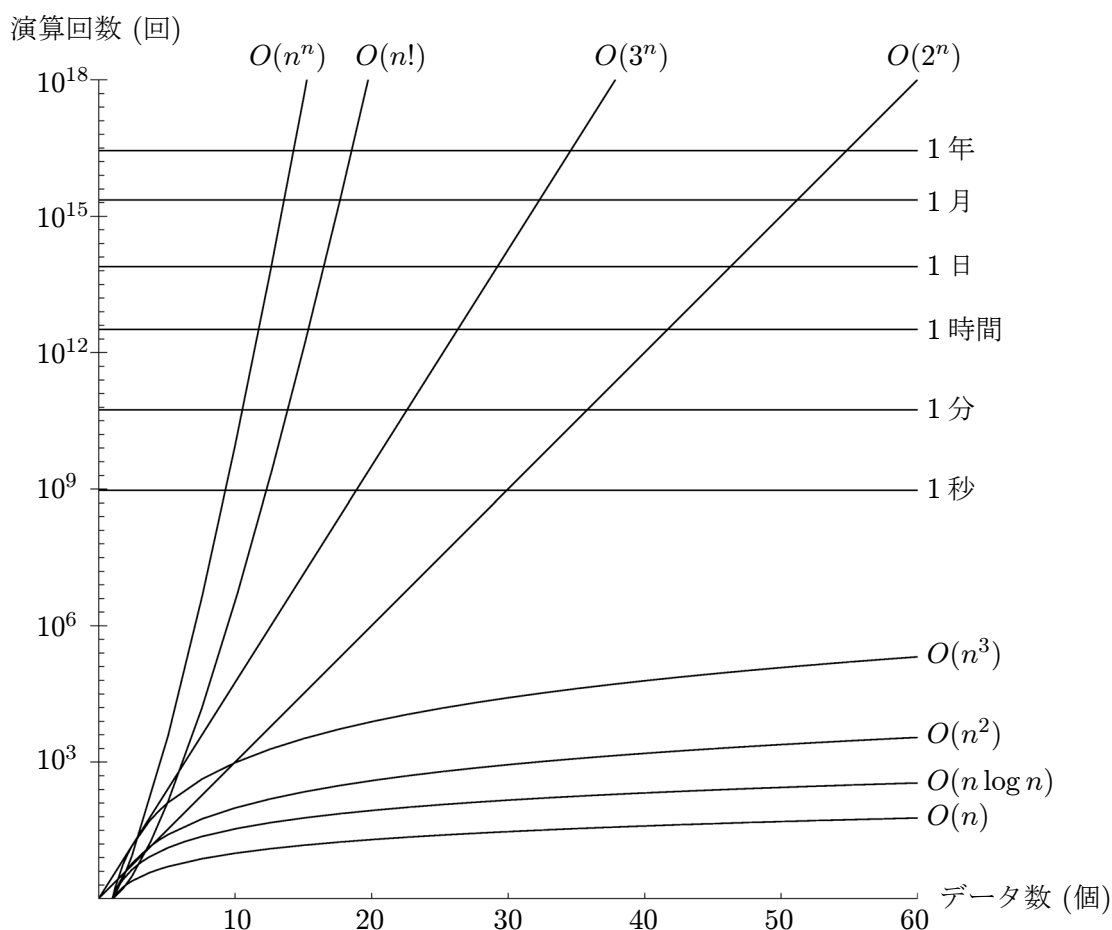


表 5.2: 計算量による評価