3　クラス $\mathcal{P}$，$\mathcal{NP}$, $\mathcal{NP}$ 完全，$\mathcal{NP}$ 困難

最悪計算量解析によって計算効率を評価することはできたが，解析された計算量の何を基準に「好ましいアルゴリズム」と「好ましくないアルゴリズム」を区別すればよいのだろうか．

3.1　多項式時間アルゴリズム vs 指数時間アルゴリズム

アルゴリズムの最悪計算量が問題の入力ビット長の多項式関数で抑えられるとき，そのアルゴリズムを**多項式時間アルゴリズム** (polynomial-time algorithm) という．ナップサック問題であれば，

$$\log b,\ \ \log a_j,\ \ \log c_j,\quad j \in N$$

に各データの区切りを示すキャラクタの数を加えたものが入力ビット長となるが，$A = \max\{a_j \mid j \in N\}$, $C = \max\{c_j \mid j \in N\}$ とおけば，

$$\log b + n(\log A + \log C)$$

程度の入力ビット長をもつと考えることができる．したがって，n と $\log b$, $\log A$, $\log C$ の多項式で最悪計算量が抑えられるなら，そのアルゴリズムは多項式時間アルゴリズムである．さらに，これらのパラメータのうちでデータの個数を表すもの，ナップサック問題ならば n，の多項式で計算量が抑えられるアルゴリズムを**強多項式時間アルゴリズム** (strongly polynomial-time algorithm) という．

最短路問題 (2.3) に動的計画法を適用した LABEL_CORRECTING の最悪計算量は，入力グラフの枝数 m と頂点数 n の多項式 $O(mn)$ であり，したがって LABEL_CORRECTING は (2.3) に対する強多項式時間アルゴリズムである．一方，ナップサック問題 (2.1) を解く動的計画法の場合，最悪計算量 $O(nb)$ は品物の数 n とナップサックの容積 b の多項式ではあるが，b をその入力ビット長 $\log b$ の関数として明示すると $O(n2^{\log b})$ となってしまう．このことは，動的計画法の最悪計算量がナップサック問題の入力ビット長を表す n, $\log b$, $\log z$, $\log A$, $\log C$ の多項式にならないことを意味している．

最悪計算量が問題の入力ビット長の多項式では抑えられないアルゴリズムは**指数時間アルゴリズム** (exponential-time algorithm) とよぶ．つまり，ナップサック問題 (2.1) に対する動的計画法は指数時間アルゴリズムである．なお，後で説明するように，現段階でナップサック問題を多項式時間で解決するアルゴリズムの存在は期待できそうにない．

最悪計算量解析に基づく 2 種類の実行時間のうち，多項式時間アルゴリズムと指数時間アルゴリズムのどちらが好ましいかは，100MIPS(1 秒間に 1 億回の基本演算が可能) の計算機を想定した次の結果から一目瞭然であろう:

入力 n	実行時間 $n \log n$	n^2	n^5	2^n	$n^2 2^n$	$n!$
10	3.32×10^{-7} 秒	1×10^{-6} 秒	0.001 秒	1.02×10^{-5} 秒	0.001 秒	0.036 秒
20	8.64×10^{-7} 秒	4×10^{-6} 秒	0.032 秒	1.05×10^{-2} 秒	4.19 秒	771 年
30	1.47×10^{-6} 秒	9×10^{-6} 秒	0.243 秒	10.7 秒	2.68 時間	5.61×10^{6} 宇宙年
40	2.13×10^{-6} 秒	1.6×10^{-5} 秒	1.02 秒	3.05 時間	204 日	1.72×10^{22} 宇宙年
50	2.82×10^{-6} 秒	2.5×10^{-5} 秒	3.13 秒	103 日	8.93 世紀	6.42×10^{38} 宇宙年

「宇宙年」は，ビックバンが起きてから現在までの約 150 億年を意味するが，ここだけの造語なので他所では使わないこと！ 入力ビット長 n の増加とともにいずれの実行時間も増加はするが，$n \log n$, n^2, n^5 などの多項式時間の基本演算は増加の速さが大したものでないことがわかる．これに対して，2^n, $n^2 2^n$, $n!$ などの指数時間の演算が行われると，非常に小さな n のときには多項式時間の演算と遜色ないものの，そののち実行時間は急激に増大し，$n = 50$ 程度の入力に対してすら宇宙の終わりまでに答が得られるかどうか怪しい．

3.2 認識問題と $\mathcal{NP}$, $\mathcal{NP}$ 完全

実は，最適化問題はもとより，アルゴリズムを使って解かれる各種の問題の中で多項式時間に解決可能な問題はむしろ希であり，多くはその可能性すら絶望視されている．そのような問題の代表が，先の最適化ナップサックであり，次に示すハミルトン閉路問題である．

ハミルトン閉路

入力: グラフ $G = (V,\ E)$

質問: グラフ G にハミルトン閉路は存在するか？

ハミルトン閉路 (Hamiltonian circuit) とは，グラフ G のすべての頂点 $i \in V$ を 1 回だけ通る閉路をさし，ハミルトン閉路問題は，与えられたグラフ G にそうした閉路が含まれているかどうかを尋ねる単純な問題である．ところが，この問題や先の最適化ナップサックには，多項式時間アルゴリズムなど存在しないというのが大方の研究者に共通の見解である．その理論的な裏付けとなっているのが，これから説明する $\mathcal{NP}$ **完全** ($\mathcal{NP}$-complete) の概念である．

クラス $\mathcal{NP}$

ハミルトン閉路問題など，次の特徴をもつ**認識問題** (recognition problem) — “yes” か “no” で答えられる問題 — のクラスを Nondeterministic Polynomial-time solvable の頭文字をとって $\mathcal{NP}$ とよぶ．

(a) 答の証拠を教えてもらえば，それが問題の条件を満たすかどうかのチェックを多項式時間で行える．

例えば，ハミルトン閉路問題に対し，答の証拠として $n = |V|$ 個の頂点の配列 $(i_1,\ i_2,\ \ldots,\ i_n)$ が与えられれば，

$$(i_k,\ i_{k+1}) \in E, \quad k = 1,\ 2,\ \ldots,\ n-1;\ \ (i_n,\ i_1) \in E$$

が満たされるかどうか，そして $(i_1,\ i_2,\ \ldots,\ i_n)$ をソートして $(1,\ 2,\ \ldots,\ n)$ に戻るかどうかを確かめればよく，明らかに n と $m = |E|$ の多項式時間で処理できる．したがって，ハミルトン閉路問題は $\mathcal{NP}$ に属する．

クラス $\mathcal{NP}$ 完全

$\mathcal{NP}$ 問題を「解く」には，しかし，

(b) 答の候補の数が入力ビット長の指数関数だけあり，それらの証拠を 1 つ 1 つチェックするという素朴な方法では，結局，指数時間が必要となる

ような可能性もある．実際，ハミルトン閉路問題では，答の候補の証拠として単純に頂点の順列 $(i_1,\ i_2,\ \ldots,\ i_n)$ を列挙すると $n!$ 個にもおよび，何らかの方法で候補を絞り込まないかぎり，(b) のような状況が起こることは避けられない．どんな方法で候補を絞り込もうとも (b) の可能性を否定しきれないとき，計算の複雑さの理論が用いるロジックは，

> 「この問題 A が多項式時間で解けるくらいなら，あの難問 B だって多項式時間で解ける」(したがって，A はたぶん多項式時間では解けない)

といういささか消極的なものである．これを厳密に定義すると次のようになる．

多項式時間還元 (polynomial-time reduction)**:** 問題 A を 1 回の基本演算と同じ時間で解くアルゴリズム α があれば，α をサブルーチンとして用いることで問題 B を多項式時間で解くことができるとき，B は A に多項式時間還元可能であるという．

この定義を使えば，(b) の可能性のある $\mathcal{NP}$ 問題を次のように分類できる．

$\mathcal{NP}$ **完全** ($\mathcal{NP}$-complete)**:** 問題 A が $\mathcal{NP}$ で，すべての $\mathcal{NP}$ 問題が A へ多項式時間還元可能なとき，A は $\mathcal{NP}$ 完全であるという．

言い換えれば，$\mathcal{NP}$ 完全な問題はどんな $\mathcal{NP}$ 問題と比べても**それ以上に**難しい．ところが $\mathcal{NP}$ 問題は，ハミルトン閉路だけでなく，星の数ほど存在するので，いま考えてる問題 A が $\mathcal{NP}$ 完全かどうかを，**すべての** $\mathcal{NP}$ 問題を A に多項式時間還元することによって確かめられようはずなどない．そこで役に立つのが次の定理で，これによって他の $\mathcal{NP}$ 完全問題をハミルトン閉路から芋ヅル式に明らかにすることができる．

定理 3.1. 問題 A が $\mathcal{NP}$ で，問題 B を任意の $\mathcal{NP}$ 完全問題とするとき，B が A へ多項式時間還元可能ならば，A は $\mathcal{NP}$ 完全である．

ハミルトン閉路問題はすでに $\mathcal{NP}$ 完全であることが知られているが，ここで試しに定理 3.1 を使って，次の問題へハミルトン閉路が多項式時間で還元できることを示し，この問題もまた $\mathcal{NP}$ 完全であることを証明してみよう．

巡回セールスマン

入力: 都市の有限集合 $C = \{1, 2, \ldots, n\}$，任意の 2 都市 $i,\ j \in C$ 間の距離 $d(i,j) \in \mathbb{Z}_+$，および移動距離の上限 $b \in \mathbb{Z}_+$

質問: 集合 C の都市すべてを 1 度だけ通り，移動距離が b を越えない旅程，$\sum_{k=1}^{n-1} d(i_k, i_{k+1}) + d(i_n, i_1) \leq b$ を満たす $\{1, 2, \ldots, n\}$ の順列 $(i_1, i_2, \ldots, i_n)$ は存在するか？

では多項式時間還元の定義にしたがって，巡回セールスマンを解く架空のアルゴリズム α をサブルーチンに用いるハミルトン閉路のアルゴリズム β を作成しよう．

アルゴリズム β

ステップ 1. ハミルトン閉路の問題例の入力 $G = (V,\ E)$ を読み込む．

ステップ 2. 入力のグラフ $G = (V,\ E)$ をもとに，すべての頂点対 $i,\ j \in V$ に対して

$$d(i,j) = \begin{cases} 1, & (i,j) \in E \text{ の場合} \\ 2, & (i,j) \notin E \text{ の場合} \end{cases}$$

によって $i,\ j$ 間の距離を定義する．

ステップ 3. アルゴリズム α に，$C = V, d(i,j)$，$b = n$ を入力し，$\sum_{k=1}^{n-1} d(i_k, i_{k+1}) + d(i_n, i_1) \leq b$ を満たす順列 $(i_1, i_2, \ldots, i_n)$ が存在するかどうかを尋ねる．

ステップ 4. アルゴリズム α が “yes” を返せば “yes”，“no” を返せば “no” を出力して終了．

ハミルトン閉路には頂点の数 n と同じ数の枝しか含まれないので，このアルゴリズム β のステップ 2 で行われる $d(i,j)$ の定義から直ちに

$$\text{順列 } (i_1, i_2, \ldots, i_n) \text{ が } \sum_{k=1}^{n-1} d(i_k, i_{k+1}) + d(i_n, i_1) \leq n \text{ を満たす} \iff (i_1, i_2, \ldots, i_n) \subset G$$

であることがわかる．つまり，巡回セールスマンを解くアルゴリズム α さえあれば，それを使ってハミルトン閉路問題も解けてしまう．

次にアルゴリズム β の最悪計算量を考えてみよう：

ステップ 1 の手間． グラフ G の各枝がどの頂点から頂点へ接続しているかさえわかればよいので，$m = |E|$ に比例する時間しか掛からない．

ステップ 2 の手間． すべての頂点対 i, j に対して距離 $d(i,j)$ を定義し，それぞれに 1 か 2 の長さを与えるだけ．つまり，頂点対の総数 $n(n-1)$ / 2 に比例する時間が掛かるにすぎない．

ステップ 3 の手間． 多項式時間還元の仮定により，アルゴリズム α は 1 回の基本演算と同じ時間しか必要としない．

ステップ 4 の手間: “yes” か “no” を出力するだけなので定数時間で済む．

以上から，アルゴリズム β は全体で $O(n^2 + m)$ の計算量しか必要としないことがわかる．つまり,「ハミルトン閉路問題は巡回セールスマンへ多項式時間還元可能」であり，定理 3.1 から巡回セールスマンはハミルトン閉路と同様，$\mathcal{NP}$ 完全問題であることが証明された．巡回セールスマンがクラス $\mathcal{NP}$ に属するかの議論を省いてしまったが，その答の証拠として旅程を表す順列 $(i_1, i_2, \ldots, i_n)$ が与えられれば，n に比例する手間で $\sum_{k=1}^{n-1} d(i_k, j_{k+1}) + d(i_n, i_1) \leq b$ をチェックできることは明らかであろう．

3.3 最適化問題と $\mathcal{NP}$ 困難

上で作成したハミルトン閉路を解くアルゴリズム β のサブルーチンとして，巡回セールスマンのアルゴリズム α を呼ぶ代わりに，次の最適化問題を解くアルゴリズム α' を呼べばどうなるだろうか？

最適化巡回セールスマン

入力: 都市の有限集合 $C = \{1, 2, \ldots, n\}$，任意の 2 都市 i, $j \in C$ 間の距離 $d(i,j) \in \mathbb{Z}_+$

出力: 集合 C の都市すべてを 1 度だけ巡る旅程で最も短いもの，つまり $\sum_{k=1}^{n-1} d(i_k, i_{k+1}) + d(i_n, i_1)$ が最小となる $\{1, 2, \ldots, n\}$ の順列 $(i_1, i_2, \ldots, i_n)$

アルゴリズム α' が出力する旅程 $(i_1, i_2, \ldots, i_n)$ は最小の距離をもつので，$\sum_{k=1}^{n-1} d(i_k, i_{k+1}) + d(i_n, i_1) \leq n$ ならば “yes”，そうでなければ “no” をアルゴリズム β の出力とすれば良さそうである．アルゴリズム α' も単位時間しか必要としないことを仮定すれば，この修正によって β が

指数時間を要することはないので,「ハミルトン閉路は最適化巡回セールスマンへ多項式時間還元可能」といえる．それでは「最適化巡回セールスマンも $\mathcal{NP}$ 完全か？」と問われれば，答は "no" である．

$\mathcal{NP}$ **困難** ($\mathcal{NP}$-hard)**:** すべての $\mathcal{NP}$ 問題が A へ多項式時間還元可能なとき，A は $\mathcal{NP}$ 困難であるという．

$\mathcal{NP}$ 完全の定義と比べると,「問題 A が $\mathcal{NP}$ で」という部分が抜け落ちている．最適化巡回セールスマンは，そもそも "yes" や "no" で答えられる認識問題ではなく，したがって $\mathcal{NP}$ に属していない．同様に，最適化ナップサックへは次の $\mathcal{NP}$ 完全問題を多項式時間で還元できるが，最適化ナップサックも認識問題ではなく，したがって $\mathcal{NP}$ 困難である．

ナップサック

入力: 品物の集合 N，各品物 $j \in N$ の大きさ $a_j \in \mathbb{Z}_+$，価値 $c_j \in \mathbb{Z}_+$，およびナップサックの容積 $b \in \mathbb{Z}_+$ と価値の和の目標値 $z \in \mathbb{Z}_+$

質問: $\sum_{j \in M} a_j \leq b$ と $\sum_{j \in M} c_j \geq z$ を同時に満たす N の部分集合 $M \subset N$ は存在するか?

しかし，これらの問題は任意の $\mathcal{NP}$ 完全問題から多項式時間で還元できるので，どんな $\mathcal{NP}$ 問題と比べてもそれ以上に難しい．そこで，最適化巡回セールスマンや最適化ナップサックのような問題は $\mathcal{NP}$ 困難と総称されているわけである．

問 3.1. 認識問題ナップサックが最適化ナップサックに多項式時間で還元できることを示せ．

3.4 クラス $\mathcal{P}$ と「$\mathcal{P} \neq \mathcal{NP}$」予想

$\mathcal{NP}$ 完全に対し，与えられた証拠のチェックだけでなく，答までも多項式時間で求められる認識問題のクラスを，Polynomial-time solvable の頭文字を取って $\mathcal{P}$ とよぶ．ハミルトン閉路問題やナップサックに多項式時間アルゴリズムが絶望視されている理由は，一般に $\mathcal{P} \neq \mathcal{NP}$ が信じられているからに過ぎない．$\mathcal{NP}$ 完全問題が 1 題でも多項式時間で解けることを証明できれば，定理 3.1 から直ちにこの不等号は否定されるが，それに成功した研究者はまだ 1 人もいない．

卵が先か，ニワトリが先か

上に示したとおり，巡回セールスマンはハミルトン閉路問題から多項式時間で還元できるので $\mathcal{NP}$ 完全である．それならば，ハミルトン閉路の $\mathcal{NP}$ 完全性はどうやって証明されたのであろうか？ — それは，巡回セールスマンの $\mathcal{NP}$ 完全性が認められたので，そこからハミルトン閉路

に多項式時間で還元される — ことはもちろんできるが,「卵が先か，ニワトリが先か」の議論に陥ってしまう．実は，ハミルトン閉路は 1972 年に R.M. Karp によって次の $\mathcal{NP}$ 完全問題から多項式時間還元されている．

頂点被覆

入力:　グラフ $G = (V,\ E)$ と整数 $k \leq |V|$

質問:　グラフ G の大きさが k 以下の頂点被覆 — $|V'| \leq k$ で各枝 $(i,\ j) \in E$ に対して頂点 i と j の少なくとも一方を含む頂点集合 $V' \subseteq V$ — は存在するか？

それでは，この頂点被覆は？ といえば，同じく Karp が次の $\mathcal{NP}$ 完全問題から多項式時間還元している．

3 充足性

入力:　論理変数 $u_1,\ \ldots,\ u_n$ と高々3 個の論理変数の和からなる項 $C_1,\ \ldots,\ C_m$

質問:　$C_1 \wedge \cdots \wedge C_m = 1$ を満たす $u_1,\ \ldots,\ u_n$ は存在するか？

それでは，3 充足性は？ といえば，S.A. Cook が次の $\mathcal{NP}$ 完全問題から多項式時間還元している．

充足性

入力:　論理変数 $u_1,\ \ldots,\ u_n$ と項 $C_1,\ \ldots,\ C_m$

質問:　$C_1 \wedge \cdots \wedge C_m = 1$ を満たす $u_1,\ \ldots,\ u_n$ は存在するか？

それでは，充足性は？ といえば，— $\mathcal{NP}$ 完全問題の記念すべき題 1 号で，1971 年に Cook は定理 3.1 の助けを借りずに（何しろ，充足性問題へ還元すべき $\mathcal{NP}$ 完全問題はそれまで発見されていない！）次の命題を証明している．

定理 3.2.　[**Cook の定理**]
任意の $\mathcal{NP}$ 問題は充足性に多項式時間還元可能である．

宿題

1 題の $\mathcal{NP}$ 完全問題が多項式時間で解ければ,「$\mathcal{P} = \mathcal{NP}$」が成り立つことを証明せよ.

ヒント: その 1 題の $\mathcal{NP}$ 完全問題を A とすれば, 任意の $\mathcal{NP}$ 完全問題 B は以下の手順で解くことができる:

ステップ 1. 問題 B の入力の読み込み.

ステップ 2. 以下を繰り返す:

(i) 問題 B の入力, アルゴリズム α の出力をもとに問題 A の入力を生成.

(ii) 問題 A を解くアルゴリズム α の呼び出し.

ステップ 3. “yes”, “no” の出力.

アルゴリズム α の必要とする手間が $O(1)$ ならば, これらのステップに掛かる手間は問題 B の入力ビット長 n の多項式 $f(n)$ で抑えられることに注意 (多項式還元の定義). ステップ 2 (i) における問題 A の入力生成の手間も $f(n)$ で抑えられる. つまり, ステップ 2 (ii) で解かれる A の入力ビット長も $f(n)$ で抑えられなければならない. また, 多項式時間還元の定義ではアルゴリズム α の呼び出しが繰り返されることは禁じられていないが, アルゴリズム全体が $f(n)$ で抑えれれているので, ステップ 2 の反復回数も $f(n)$ を上限としなければならない.

$\mathcal{NP}$ 完全に関する詳細と宿題の答は以下の文献に解説されているので, わからないときにはこれらを参照のこと:

[1] Garey, M.R. and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W.H. Reeman, 1979).

[2] Papadimitriou, C.H. and Steiglitz, K.S., *Combinatorial Optimization: Algorithms and Complexity* (Prentice Hall, 1982).